

AD-A135 442

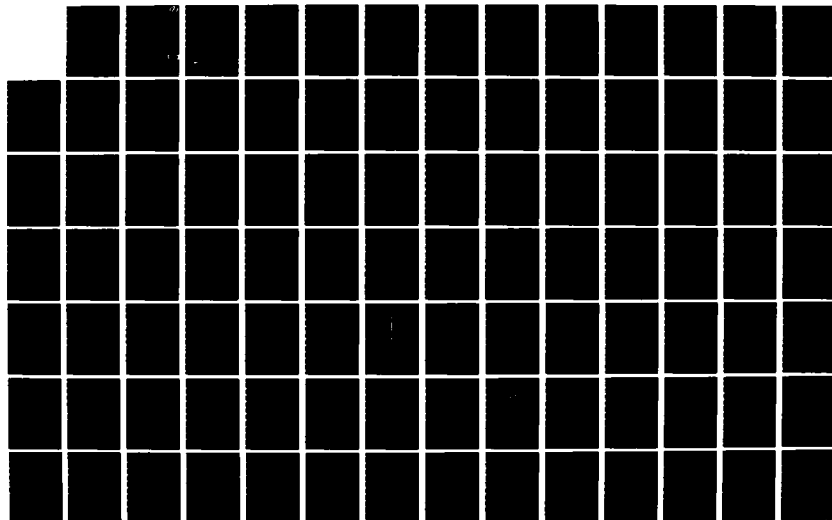
SELECTION OF NOISY SENSORS AND ACTUATORS FOR REGULATION
OF LINEAR SYSTEMS(U) AIR FORCE INST OF TECH
WRIGHT-PATTERSON AFB OH M L DELORENZO AUG 83
AFIT/CI/NR-83-59D

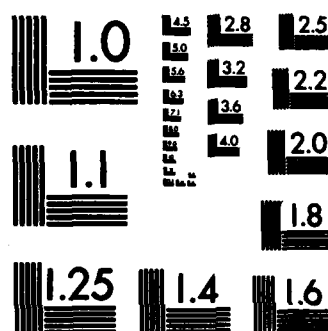
1/3

UNCLASSIFIED

F/G 12/1

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD-A135-442

①

SELECTION OF NOISY SENSORS AND ACTUATORS
FOR REGULATION OF LINEAR SYSTEMS

A Thesis

Submitted to the Faculty

of

Purdue University

by

Michael L. DeLorenzo

In Partial Fulfillment of the
Requirements for the Degree

of

Doctor of Philosophy

August 1983

DTIC
ELECTE
DEC 7 1983
S D

DTIC FILE COPY

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

00 12 00 095

UNCLASS

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFIT/CI/NR 83-59D	2. GOVT ACCESSION NO. AN 413542	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Selection Of Noisy Sensors And Actuators For Regulation Of Linear Systems		5. TYPE OF REPORT & PERIOD COVERED THESIS/DISSERTATION
7. AUTHOR(s) Michael L. DeLorenzo		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS AFIT STUDENT AT: Purdue University		8. CONTRACT OR GRANT NUMBER(s)
11. CONTROLLING OFFICE NAME AND ADDRESS AFIT/NR WPAFB OH 45433		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE August 1983
		13. NUMBER OF PAGES 242
		15. SECURITY CLASS. (of this report) UNCLASS
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES APPROVED FOR PUBLIC RELEASE: IAW AFR 190-17 29 NOV 83 Lynn E. Wolaver Dean for Research and Professional Development		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)		Accession For NTIS GRA&I <input checked="" type="checkbox"/> DTIC TAB <input type="checkbox"/> Unannounced <input type="checkbox"/> Justification
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) [REDACTED]		By Distribution/ Availability Codes Avail and/or Special Dist A/I

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 83 IS OBSOLETE

UNCLASS

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

To my wife Cathy, and my daughters Elli and Sarah. Their sacrifice for me has helped me better understand God's love for me!

ACKNOWLEDGMENTS

I would like to thank Professor Robert E. Skelton for his help in focusing the scope of this research. In addition, his knowledge of control theory coupled with his personal commitment to me served as one of the major factors in the completion of this research. The time commitment and helpful suggestions of the other members of my committee, Professor Henry T. Yang, Professor Leonard D. Berkovitz, and Professor Dominick Andrisani II is also deeply appreciated.

This research involved a substantial amount of computer work. It is *impossible* to estimate how many hours were saved as a result of the help and advice of my good friend Captain James A. Davis. I would also like to thank my typist, Ms. Terri Denison, for her quality typing, her extreme patience, and her big smile in the face of less than sane time requirements. Last, but most certainly *not* least, I would like to acknowledge the prayers of my Christian brothers and sisters (in which my family is included) that kept God's peace continually around me.

TABLE OF CONTENTS

	Page
LIST OF TABLES	vii
LIST OF FIGURES	ix
ABSTRACT	xi
 1.0 INTRODUCTION	 1
1.1 System Model and Specifications	3
1.1.1 The Model	4
1.1.2 The Specifications	6
1.2 The SAS Problem	8
1.3 General Objective and Approach	9
1.4 Organization	10
 2.0 PAST APPROACHES TO SAS	 12
2.1 The DPM	12
2.2 SAS Criteria and General Approach	13
2.3 Literature Survey	15
2.3.1 SAS Based on a Quadratic Cost Functional	15
2.3.2 SAS Based on the Error Covariance Matrix	19
2.3.3 Other SAS Criterion	23
2.4 General Relation to the SAS Problem	28
 3.0 LQG THEORY	 30
3.1 The Steady State LQG Controller	30
3.2 LQG Theory and the SAS Problem	33
3.3 LQG Theory Advantages and Disadvantages	37
 4.0 THE LARGE SPACE STRUCTURE (LSS)	 39
4.1 General LSS Description and Mission	39
4.2 The LSS Model	40
4.3 The Hoop Column Antenna	48
4.3.1 Hoop Column NASTRAN Data and Model	51
4.3.2 The Hoop Column SASLQG Problem	57

	Page
4.4 The Solar Optical Telescope.	64
4.4.1 Telescope Model Development	67
4.4.2 The Telescope SASLQG Problem.	74
5.0 ACTUATOR AND SENSOR EFFECTIVENESS VALUES.	75
5.1 Closed-loop Input and Output Cost Analysis	75
5.2 Derivation of Effectiveness Values	79
5.2.1 Actuator Effectiveness Value (V_i^{act})	80
5.2.2 Sensor Effectiveness Value (V_i^{sen})	86
6.0 LQG WEIGHT SELECTION.	92
6.1 Past Approaches to Q and R Selection	94
6.2 Input/Output-Constrained Solutions	98
6.2.1 Input-Constrained Discussion ($S(n,1,1,\ell)$)	98
6.2.2 Output-Constrained Discussion ($S(n,1,1,\ell)$)	99
6.2.3 Multi-Input/Output-Constrained Solutions.	100
6.3 Theory	103
6.3.1 Definitions	103
6.3.2 Theorems.	104
6.3.3 Update Equations.	108
6.3.3.1 Input Update Equation.	108
6.3.3.2 Output Update Equation	111
6.3.3.3 Simultaneous Input and Output Updates.	113
6.3.4 Existence of R_M , Q_M , Q_M^* , R_M^*	114
6.3.4.1 Existence of R_M	114
6.3.4.2 Existence of Q_M	115
6.3.5 Summary	116
6.4 The Algorithm LQGWS	116
6.5 Hoop Column Antenna Example.	120
6.5.1 Input-Constrained Solution.	120
6.5.2 Output-Constrained Solution	126
6.5.3 Summary of Hoop Column Results.	136
6.6 Solar Optical Telescope Example.	136
6.6.1 Input-Constrained Solution.	137
6.6.2 Output-Constrained Solution	137
6.6.3 Summary of Telescope Results.	146
7.0 THE SASLQG ALGORITHM.	148
7.1 Important Facts and Basic Algorithm Flow	150
7.2 The Algorithm.	154
7.3 Hoop Column SASLQG Example	156
7.3.1 Input-Constrained and Output-Constrained SASLQG Results.	157
7.3.2 Sensor and Actuator Configuration Comparisons	161

	Page
7.3.3 Physical Insights.164
7.4 Telescope SASLQG Example.168
7.4.1 Input-Constrained and Output-Constrained Results.168
7.4.2 Sensor and Actuator Configuration Comparisons. . .	.172
7.4.3 Physical Insights.174
7.5 Concluding Comments174
8.0 CONCLUSION177
8.1 Contributions178
8.2 Recommendations for Further Research.178
8.2.1 Discrete Time Systems/Deterministic Systems. . .	.179
8.2.2 Sensors Constraints.179
8.2.3 SASLQG Simplification.179
8.2.4 Non-Diagonal Q and R180
8.2.5 PWR(j) and PWR _y (j)180
LIST OF REFERENCES.181
APPENDICES	
Appendix A: Proofs.187
Appendix B: V_i^V and the Chen-Seinfeld Switching-Function . .	.204
Appendix C: Conservative Step Size Argument206
Appendix D: Necessary and Sufficient Conditions for $E_{\infty} u_i^2 = 0 \quad \forall Q, R > 0$210
Appendix E: Program Listings.214
VITA.242

LIST OF TABLES

Table	Page
4.1: Hoop Column Output Description.49
4.2: Hoop Column Sensor Description.50
4.3: Hoop Column Actuator Description.51
4.4: Multi-dimensional $\Omega(t,s)$ notation52
4.5: Partial Representation of \tilde{C}_p , \tilde{M}_p , \tilde{M}_R , and f54
4.6: The BE and BR Matrices.58
4.7: The CE Matrix59
4.8: The CR Matrix60
4.9: The ME Matrix61
4.10: The MR Matrix62
4.11: The MER and MRR Matrices.63
4.12: Telescope Sensor Description.65
4.13: Telescope Actuator Description.66
4.14: Telescope Specifications.66
4.15: The S and BE Matrices70
4.16: The BR, CE and CR Matrices.71
4.17: The ME Matrix72
4.18: The MR Matrix73
5.1: Hoop Column Weighting Matrices, 182
5.2: Hoop Column Weighting Matrices, 284
5.3: Telescope Weighting Matrices85

Table	Page
6.1: Hoop Column Input-Constrained Solution.	121
6.2: Hoop Column Output-Constrained Solution	128
6.3: Telescope Input-Constrained Solution.	138
6.4: Telescope Output-Constrained Solution	139
7.1: Hoop Column Sensor Labels	157
7.2: Hoop Column Input-Constrained SASLQG Results.	159
7.3: Hoop Column Output-Constrained SASLQG Results	160
7.4: Input-Constrained Configuration Comparison.	162
7.5: Output-Constrained Configuration Comparison	163
7.6: H.SAS Input-Constrained Specifications.	165
7.7: H.SAS Output-Constrained Specifications	166
7.8: Telescope Sensor Labels	169
7.9: Telescope Input-Constrained SASLQG Results.	170
7.10: Telescope Output-Constrained SASLQG Results	171
7.11: Input/Output-Constrained Configuration Comparison (T.SAS)	173
7.12: T.SAS Input-Constrained Specifications.	175
7.13: T.SAS Output-Constrained Specifications	175

LIST OF FIGURES

Figure	Page
4.1: Hoop Column Antenna.48
4.2: The Solar Optical Telescope.64
5.1: Hoop Column v_i^{act} Data for Q and R of Table 5.1.83
5.2: Hoop Column v_i^{act} Data for Q and R of Table 5.2.84
5.3: Telescope v_i^{act} Data for Q and R of Table 5.3.85
5.4: Hoop Column v_i^{sen} Data for Q and R of Table 5.1.89
5.5: Hoop Column v_i^{sen} Data for Q and R of Table 5.2.89
5.6: Telescope v_i^{sen} Data for Q and R of Table 5.3.90
6.1: $S(n,1,1,x)$ Input-Constrained Solution.99
6.2: $S(n,1,1,x)$ Output-Constrained Solution.	100
6.3: $S(n,2,1,x)$ Input-Constrained Solution.	101
6.4: Hoop Column Input-Constrained Solution (actuators 1-4) . .	123
6.5: Hoop Column Input-Constrained Solution (actuators 5-8) . .	124
6.6: Hoop Column Input-Constrained Solution (actuators 9-12). .	125
6.7: Hoop Column Input-Constrained Solution (outputs 13-16) . .	127
6.8: Hoop Column Output-Constrained Solution (outputs 1-4). . .	129
6.9: Hoop Column Output-Constrained Solution (outputs 5-8). . .	130
6.10: Hoop Column Output-Constrained Solution (outputs 9-12) . .	131
6.11: Hoop Column Output-Constrained Solution (outputs 13-16). .	132
6.12: Hoop Column Output-Constrained Solution (outputs 17-20). .	133

Figure	Page
6.13: Hoop Column Output-Constrained Solution (outputs 21-24). . .	.134
6.14: Hoop Column Output-Constrained Solution (actuators 1-4). . .	.135
6.15: Telescope Input-Constrained Solution (actuators 1-4)140
6.16: Telescope Input-Constrained Solution (actuators 5-8)141
6.17: Telescope Input-Constrained Solution (actuators 9-12). . .	.142
6.18: Telescope Input-Constrained Solution (actuators 13-16) . .	.143
6.19: Telescope Input-Constrained Solution (actuators 17-20) . .	.144
6.20: Input-Constrained Solution (actuator 21, outputs 1-3). . .	.145
6.21: Telescope Output-Constrained Solution (actuator 21, outputs 1-3)147

ABSTRACT

DeLorenzo, Michael L. Ph.D., Purdue University, August 1983. Selection of Noisy Sensors and Actuators for Regulation of Linear Systems. Major Professor: Robert E. Skelton.

→ This research has developed and tested an algorithm which aids the controls engineer in placing sensors and actuators in a linear system to best achieve a set of variance specifications on the outputs and inputs of the system. The term best achieve has been defined to be the sensor and actuator configuration which enables a controller to do either of the following: Meet the input specifications while minimizing a sum of output variances normalized by their specification (i.e. input-constrained solution), or meet the output specifications while minimizing a sum of input variances normalized by their specification (i.e., output-constrained solution). *(Linear Quadratic Gaussian)*

The approach taken to solve this sensor and actuator selection (SAS) problem was to use LQG theory to specify a structure for the controller, and then develop an algorithm (SASLQG) that places sensors and actuators in this controller structure to achieve either the input-constrained or output-constrained solution. The main advantage of this approach is the mathematical ease with which LQG theory addresses variance constraints, and the main disadvantage is that there may be other controller structures which do better. ←

In applying LQG theory to solve the SAS problem two specific extensions of the theory resulted. The first was development of sensor and actuator effectiveness values (V_i^{sen} and V_i^{act}) which determine the importance of each sensor and actuator to the LQG controller when *both* the sensors and actuators are assumed noisy. The second extension was the development of the algorithm LQGWTS which provides a systematic method for adjusting the weighting matrices in the LQG cost functional V so that the controller which minimizes V also satisfies either the input-constrained or output-constrained variance requirements.

These two extensions were combined to form a sensor and actuator selection algorithm (SASLQG). The algorithm was applied to two substantial models of large space structures and the resulting configurations although not guaranteed to be optimal achieved better performance than any alternative configuration tested. The algorithm also provides insight into the sensitivity of the controller design to sensor and actuator deletions and therefore, insight into an optimal number for both sensors and actuators. Lastly, the algorithm provides information which identifies the most demanding outputs and the critical actuators for the final sensor and actuator configuration.

1.0 INTRODUCTION

Our ability to make the behavior of a physical system (human or not) conform to an unnatural but necessary standard (i.e. regulate it) is directly proportional to our influence on the system, our understanding of how the system responds to this influence, our perception of current system behavior, the way we exert our influence (friendly persuasion, brute force etc.), and of course the severity of the necessary standard. From an engineering perspective, the standard of system performance is usually defined by a set of specifications (constraints) on system output(s) and/or input(s). Systems normally receive inputs (influence) through physical devices called actuators. The understanding of how a system responds to inputs is represented, in most cases, by a set of differential equations referred to as a system model. Current system behavior is normally monitored by devices called sensors and the technique for combining sensor and model information into a set of rules for issuing actuator commands is referred to as a control or regulation law. Control laws which use sensor information are called closed-loop control laws and are, in general, less sensitive to unexpected disturbances and implementation errors than are open-loop control laws which do not use sensor information.

Basically then, the problem of regulating physical systems has five elements :

- (1) Specifications
- (2) Actuators
- (3) System Model
- (4) Sensors
- (5) Control Law

When these elements interact during regulator design, experience has shown that the following situations arise:

- (i) The mathematical models used to represent physical systems are never exactly right and sometimes the real system is offended by actuator commands based on an imitation. (i.e. our control laws are right in theory but wrong in practice).
- (ii) There are many techniques for developing control laws and the resulting regulators can have a wide range of complexity. In addition to relative complexity, each technique has other advantages and disadvantages and no one technique always does the 'best' job.
- (iii) It is sometimes impossible to meet the given set of specifications and it is often not clear what specifications *are* achievable with the given control elements.
- (iv) Actuators have physical limits to the amount of push, pull, torque, etc. that they can generate and control laws sometimes forget this. (i.e. Input constraints are a physical reality in most practical control problems).
- (v) When a large number of admissible locations exist for sensors and actuators, some locations do better than others in achieving the regulation specifications, and systematically comparing regulator designs for *every* admissible sensor

and actuator configuration is impossible. (For instance, when a design problem requires choosing 6 out of 12 actuators and 12 out of 39 sensors, there are $\sim 3.61 \times 10^{12}$ possible configurations!).

- (vi) Sensors and actuators also have unmodeled behavior (noise) which always degrades regulator performance. This noise can significantly effect the locations sought in (v) and can invalidate the design theory which states the more actuators used the better.

This research has focused on the Sensor and Actuator Selection (SAS) questions raised by situations (iii)-(vi). Before introducing these questions further a brief discussion of the assumed system model and regulation specifications is in order.

1.1 System Model and Specifications

The modeling problem of situation (i) is present in every area of control theory and is currently a very active topic of research. For the most part our system models are based on linear constant coefficient differential equations for which a great wealth of solution techniques and control theory exist. However, physical systems are by nature, non-linear and are most accurately represented by non-linear differential equations. The problem with these type of equations is that no general procedure exists for obtaining their closed form solutions, and our ability to analyze non-linear system behavior to various forms of inputs is therefore limited. This limitation carries over to the design of regulators for non-linear systems and manifests itself in the fact that the wealth of design techniques for linear systems currently has no parallel in non-linear control theory. Fortunately, most physical systems do have 'nearly' linear behavior over limited ranges of

response. The regulators which are based upon a model of this linear behavior are, for a large number of practical cases, able to keep the system operating within its linear range and therefore maintain the validity of the linear model and the designed regulator. In some situations, adding white noise processes to the linear models can account for unmodeled non-linear behavior of the system and its sensors and actuators thus enhancing the validity and range of the linear models. These models then become linear stochastic models and they are the subject of this research.

1.1.1 The Model

The specific type of linear stochastic model considered in this research uses a set of ordinary differential equations with constant coefficients driven by random processes that are at least wide-sense stationary. This type of model is called a Lumped Parameter Model (LPM) since it represents the motion of a physical system whose mass or collection of masses can be attributed (lumped) to specific points in the system. A considerable number of physical systems can be represented by an LPM. One of these is the Large Space Structure (LSS) which was chosen for the practical application of this research. The specific LSS model development is the subject of Chapter 4. Shown below is the state space form of the LPM used in this research.

$$(1.1) \quad \begin{cases} \dot{x}(t) = Ax(t) + Bu(t) + Dw(t) ; & x \in R^n, \quad u \in R^m, \quad w \in R^p \\ x(t_0) = x_0 \\ D = [B \quad D_0] \\ y(t) = Cx(t) ; & y \in R^k & \text{(system outputs)} \\ z(t) = Mx(t) + v(t) ; & z \in R^l & \text{(system measurements)} \end{cases}$$

with noise characteristics:

$$(1.2) \quad \left\{ \begin{array}{l} \overline{Ex_0} = 0 ; \quad \overline{Ew(t)} = 0 ; \quad \overline{Ev(t)} = 0 \\ E \left\{ \begin{pmatrix} x_0 \\ w(t) \\ v(t) \end{pmatrix} (x_0^T, w^T(\tau), v^T(\tau)) \right\} = \begin{bmatrix} x_0 & 0 & 0 \\ 0 & W\delta(t-\tau) & 0 \\ 0 & 0 & V\delta(t-\tau) \end{bmatrix} ; W, V > 0 \end{array} \right.$$

Where the notation R^i implies a real vector space of dimension i , E represents the expectation operator, T represents matrix transposition, δ is the Dirac delta function and $W > 0$ implies W is a positive definite matrix.

The n -dimensional vector $x(t)$ represents the state of the system, while the m -dimensional vector $u(t)$ contains the actuator signals. The system outputs which are to be regulated are defined by the k -dimensional vector $y(t)$ and the ℓ dimensional vector $z(t)$ represents the measurements (sensor information) available from the system. The white noise vector process $w(t)$ is used to represent unmodeled system behavior ($D_0 w(t)$) and unmodeled actuator behavior or noise ($Bw(t)$), while unmodeled or noisy sensor behavior is accounted for by the white noise vector process $v(t)$. The matrices A, B, C, D, M, W , and V are assumed to be time-invariant and appropriately dimensioned. It is further assumed that the matrix B has no zero columns, the matrices C and M have no zero rows and the matrices A, B, C, D , and M satisfy the following detectability and stabilizability conditions: [1]

$$(1.3) \quad \begin{array}{ll} (A, B) \quad (A, D) & \text{stabilizable} \\ (A, C) \quad (A, M) & \text{detectable} \end{array}$$

For the purposes of notation the LPM described by (1.1)-(1.3) will be identified by $S(n, k, m, \ell)$ where n is the number of states used to

represent the system, k is the number of outputs, m is the number of actuators and l is the number of sensors.

1.1.2 The Specifications

As noted earlier, the goal of the regulation process is to keep the outputs and/or inputs (actuator signals) of a physical system within some desired range. For the system $S(n,k,m,l)$ these specifications could take the following form:

$$(1.4) \begin{cases} -\sigma_i \leq y_i(t) \leq \sigma_i & \forall t > t_0 ; i = 1, \dots, k \\ -\mu_i \leq u_i(t) \leq \mu_i & \forall t > t_0 ; i = 1, \dots, m \end{cases}$$

where σ_i and μ_i are constants representing the specifications (constraints) on the i^{th} output and input respectively. Since $S(n,k,n,l)$ is driven by white noise processes, both $x(t)$ and $y(t)$ will be random vector processes. Therefore (1.4) can become a very severe requirement and in reality could never be guaranteed. From a probabilistic view, $Ey_i(t)$ might make more sense for a regulated quantity, however, we know that $y_i(t)$ can be expressed as follows: [1]

$$(1.5) \quad y_i(t) = Ce^{A(t-t_0)}x_0 + C \int_{t_0}^t e^{A(t-\tau)}Bu(\tau)d\tau + C \int_{t_0}^t e^{A(t-\tau)}Dw(\tau)d\tau$$

and, given the noise characteristics of $S(n,k,m)$ along with removing all controls (i.e. actuators turned off),

$$(1.6) \quad Ey_i(t) = 0 \quad \forall t > t_0, \quad i = 1, \dots, k$$

which produces a meaningless regulation problem.

The preceding discussion leads quite naturally to the use of the variance constraints shown in (1.7).

$$(1.7) \quad \begin{cases} E y_i^2(t) \leq \sigma_i^2 & \forall t > t_0, i = 1, \dots, k \\ \text{and} \\ E u_i^2(t) \leq \mu_i^2 & \forall t > t_0, i = 1, \dots, m \end{cases}$$

The constant quantities σ_i^2 and μ_i^2 now represent the variance constraints on the i^{th} output and input respectively. The constraint on $u_i(t)$ is necessary since only closed-loop regulation will be considered and $u_i(t)$ must then be a function of noisy sensor information which is further a function of the random vector process $x(t)$. For systems of type $S(n, k, m, \lambda)$ control laws are known to exist which produce steady state values for $E y_i^2(t)$ and $E u_i^2(t)$ and since a great many practical regulation problems require the outputs to be regulated for long periods of time the following specifications have proven to be a very desirable alternative to (1.7):

$$(1.8) \quad \begin{cases} \lim_{t \rightarrow \infty} E y_i^2(t) \triangleq E_{\infty} y_i^2 \leq \sigma_i^2, & i = 1, \dots, k \\ \text{and} \\ \lim_{t \rightarrow \infty} E u_i^2(t) \triangleq E_{\infty} u_i^2 \leq \mu_i^2, & i = 1, \dots, m \end{cases}$$

The specifications of (1.8) were used in this research and the expression (σ^2, μ^2) will be used to imply (1.8).

1.2 The SAS Problem

With the background of Section 1.1 it is now possible to formulate situations (iii)-(vi) as a specific SAS problem for systems of type $S(n, k, m, \ell)$. To begin, assume that only \bar{m} out of m actuators and $\bar{\ell}$ out of ℓ sensors are available for designing a regulator with specifications $(\sigma^2, \bar{\mu}^2)$. The goal is to find $S(n, k, \bar{m}, \bar{\ell})$ and its resulting closed-loop controller $u(\bar{\ell}, \bar{m})$ such that $(\sigma^2, \bar{\mu}^2)$ is 'best achieved', where $\bar{\mu}^2$ is the specification for the reduced set of actuators. The term 'best achieved' warrants further explanation. If it is possible to achieve the specifications $(\sigma^2, \bar{\mu}^2)$, the combination $[S(n, k, \bar{m}, \bar{\ell}), u(\bar{m}, \bar{\ell})]$ which 'best achieves' $(\sigma^2, \bar{\mu}^2)$ is defined as that combination which produces the smallest possible value for one of the following normalized sums:

$$(1.9) \quad \text{Min}_{S, u} \sum_{i=1}^k E_{\omega} y_i^2 / \sigma_i^2 \quad \text{subject to} \quad E_{\omega} u_i^2 = \bar{\mu}_i^2 \quad \forall i = 1, \dots, \bar{m}$$

or

$$(1.10) \quad \text{Min}_{S, u} \sum_{i=1}^{\bar{m}} E_{\omega} u_i^2 / \bar{\mu}_i^2 \quad \text{subject to} \quad E_{\omega} y_i^2 = \sigma_i^2 \quad \forall i = 1, \dots, k$$

If it is not possible to achieve $(\sigma^2, \bar{\mu}^2)$ the combination $[S(n, k, \bar{m}, \bar{\ell}), u(\bar{m}, \bar{\ell})]$ which 'best achieves' $(\sigma^2, \bar{\mu}^2)$ is defined to be that combination which minimizes one of the following normalized sums of outputs above specification or inputs above specification:

$$(1.11) \quad \text{Min}_{S, u} \sum_{i=1}^k E_{\omega} y_i^2 / \sigma_i^2 \quad \forall i: E_{\omega} y_i^2 > \sigma_i^2$$

subject to $E_{\omega} u_i^2 = \bar{\mu}_i^2 \quad \forall i = 1, \dots, \bar{m}$

$$\begin{aligned}
 (1.12) \quad & \text{Min}_{S,u} \sum_{i=1}^{\bar{m}} E_{\omega} u_i^2 / \bar{\mu}_i^2 \quad \forall i: E_{\omega} u_i^2 > \bar{\mu}_i^2 \\
 & \text{subject to} \quad E_{\omega} y_i^2 = \sigma_i^2 \quad \forall i = 1, \dots, k
 \end{aligned}$$

The $[S(n,k,\bar{m},\bar{\ell}), u(\bar{\ell},\bar{m})]$ which satisfies either (1.9) or (1.11) will be called the *input-constrained* SAS solution and the combination which satisfies (1.10) or (1.12) will be called the *output-constrained* SAS solution. The above discussion may be summarized in the following concise SAS problem statement:

SAS Problem Statement

Given: a system of type $S(n,k,m,\ell)$ with only \bar{m} out of m actuators and $\bar{\ell}$ out of ℓ sensors available for designing a regulator to achieve the specifications (σ^2, μ^2) .

Required: Specify the system $S(n,k,\bar{m},\bar{\ell})$ and the closed-loop controller $u(\bar{\ell}, \bar{m})$ which satisfies either the input constrained requirements of (1.9) and (1.11) or the output-constrained requirements of (1.10) and (1.12).

1.3 General Objective and Approach

The objective of this research has been to develop and test an algorithm which aids the controls engineer in finding a solution to the SAS problem. There are different ways to achieve this objective but they all must have at least the following ingredients.

- (1) A specific structure for the closed-loop control law.
- (2) Some technique for adjusting the parameters of the control law to achieve $(\sigma^2, \bar{\mu}^2)$.

- (3) Some means other than a direct search technique to evaluate the effectiveness of the various possible sensor and actuator configurations in achieving (σ^2, \bar{u}^2) .

The approach in this research was to use the well documented theory of Linear-Quadratic-Gaussian (LQG) control to specify the structure of the closed-loop controller. Using the concept of component cost analysis developed by Skelton and co-workers, [4]-[7], a technique was developed for determining the effectiveness of individual sensor and actuators in the minimization effort of the LQG cost functional. Then, a method for adjusting the weights of the cost functional to achieve (σ^2, μ^2) was developed. With this link between the cost functional and (σ^2, μ^2) established, the actuator and sensor effectiveness values were used along with the weight specification technique to develop an iterative design algorithm for the SAS problem.

1.4 Organization

The presentation of the design algorithm is organized as follows: a survey of past approaches to SAS is presented in Chapter 2, while Chapter 3 contains the specific details of LQG theory, the advantages and disadvantages for its use in solving the SAS problem and a redefinition of the research objective in terms of the mathematics of LQG theory. In Chapter 4 the two substantial LSS models used to test the design algorithm are defined. Also included in Chapter 4 is a development of the model of type $S(n,k,m,\lambda)$ from finite element data for one of the structures. The theory and development of the actuator and sensor effectiveness values are presented in Chapter 5, while the theory

and development for the weight selection technique is discussed in Chapter 6. The results of Chapters 5 and 6 are combined in Chapter 7 to develop the algorithm for solving the SAS problem. Chapter 7 also contains the results of the algorithm when applied to the LSS models of Chapter 4. The conclusion is presented in Chapter 8.

2.0 PAST APPROACHES TO SAS

Several approaches to SAS have been presented in the literature. None directly address the SAS problem as defined in Chapter 1; however, major components of the problem are addressed for both the LPM and the distributed parameter model (DPM). Before reviewing these SAS approaches, a brief discussion of the DPM is in order along with a description of the most used SAS criteria and the general approach chosen for SAS.

2.1 The DPM

A DPM is used to describe the motions of a physical system when the system mass is allowed to exist as a continuum throughout the spatial domain of the system. Therefore a DPM consists of a set of partial differential equations with independent variables both in time and space. The literature considered a linear stochastic DPM which can be written in a form that closely parallels (1.1).

$$(2.1) \quad \left\{ \begin{array}{l} \frac{\partial x(t,s)}{\partial t} = A_s x(t,s) + B(t,s)u(t) + D(t,s)w(t,s) - \begin{matrix} x \in R^n, u \in R^m, \\ w \in R^p \end{matrix} \\ x(t_0,s) = x_0(s) \\ y(t,s) = C(t,s) x(t,s) ; \quad y \in R^k \quad (\text{system outputs}) \\ z(t,s) = M(t,s) x(t,s) + v(t,s) \quad (\text{system measurements}) \end{array} \right.$$

The spatial coordinates are represented by the s dimensional real vector s which is defined on a connected open domain labeled S . A_s is defined to be a matrix differential operator (i.e. $A_s[\cdot] = A_0[\cdot] + A_1 \frac{\partial[\cdot]}{\partial s} + A_2 \frac{\partial^2[\cdot]}{\partial s^2} + \dots$) and $B(t,s)$, $D(t,s)$, $C(t,s)$ and $M(t,s)$ are known time and space dependent linear operators. As in the LPM, $v(t,s)$ and $w(t,s)$ represent white noise processes used to account for modeling inaccuracies and $x_0(s)$ is a random vector process in S which is independent of v and w .

For most physical systems it is currently not possible to generate independent control forces at every point in S or to simultaneously measure the movement of each point in S . Therefore, the literature adopted the following 'point-wise' representation for the m admissible actuators and ℓ admissible sensors of (2.1):

$$(2.1a) \quad \begin{cases} B(t,s)u(t) \triangleq \sum_{i=1}^m b_i(s) \delta(s-s_i) u_i(t) \\ z(t,s) \triangleq \sum_{j=1}^{\ell} \{M(t,s_j) x(t,s_j) + v(t,s_j)\} \end{cases}$$

Where s_i represents the spatial coordinate for the i^{th} actuator, and s_j represent the spatial coordinate for the j^{th} sensor. It should also be noted that the numerical examples presented in the literature further assumed the linear operators $B(t,s)$, $D(t,s)$, $M(t,s)$ and $C(t,s)$ to be time-invariant matrices of appropriate dimensions.

2.2 SAS Criteria and General Approach

For this selection of literature, there were two SAS criteria which were commonly used. One involved the trace of the covariance matrix

(or DPM operator) for the estimation error in the well known Kalman-Bucy filter.[1] The other was a weighted quadratic cost functional of the system state and control. Shown below are mathematical expressions of these criteria for both the LPM and DPM:

$$(2.2a) \quad \int_{t_0}^t \text{tr}\{P(t)\}dt$$

$$(2.2b) \quad \int_{t_0}^t \int_S \int_S \text{tr}\{P(t, s_1, s_2)\}ds_1 ds_2 dt$$

where $\text{tr}\{\cdot\}$ represents the trace of a matrix and

$$(2.2c) \quad P \triangleq E\{(x-\hat{x})(x-\hat{x})^T\}$$

The vector \hat{x} represents the estimate of the state vector as determined from the Kalman-Bucy filter. The quadratic criteria are:

$$(2.3a) \quad E \left[\int_{t_0}^t \{x^T(\tau)K_2x(\tau) + u^T(\tau)Ru(\tau)\}d\tau + x^T(t)K_1x(t) \right]$$

$$(2.3b) \quad E \left[\int_{t_0}^t \int_S \{x^T(\tau,s)K_2x^T(\tau,s) + u^T(\tau)Ru(\tau)\} dsd\tau + x^T(t,s)K_1x(t,s) \right]$$

where $R > 0$ and $K_2 \geq 0$, $K_1 \geq 0$ (i.e. positive semi-definite). With these criteria the goal for SAS becomes finding the set of sensors which minimizes (2.2) over all possible sets of sensors, or finding the set of actuators and/or sensors which, for a given controller structure, minimizes (2.3) over all possible sets of actuators and/or sensors.

Then general approach of the reviewed literature to SAS may now be summarized

- (1) Develop an analytical expression (criterion) which reflects some desired goal for SAS (usually minimization of (2.2) or (2.3)).
- (2) Use this criterion in a parameter optimization problem, a minimum principle formulation or a variational technique in order to derive the necessary conditions for the sensor and/or actuator configuration to achieve the SAS goal.
- (3) Use some form of gradient, successive approximation or direct search technique to select the sensor and/or actuator configuration which satisfies the necessary conditions.

2.3 Literature Survey

The literature survey will be divided into three distinct sections. The first section will summarize the techniques for selecting actuators and/or sensors to minimize (2.3). The second section will survey the literature on selecting sensors to minimize (2.2) and the last section will discuss sensor and actuator selection techniques for minimizing criteria other than (2.2) or (2.3).

2.3.1 SAS Based on a Quadratic Cost Functional

In [8], Johnson et al. propose a technique for locating a fixed number of noiseless control surfaces (actuators) on a flexible aircraft such that the controller designed to minimize an infinite time version of (2.3a) (see Chapter 3) achieves the smallest possible value of (2.3a) for the given number of control surfaces. It is assumed that the control surfaces can be located across a continuum, and the desired locations are sought by using a second order Newton-Rhapson technique to

update the actuator locations. The technique uses closed-form expressions for the first and second variations of the cost functional with respect to the control surface locations, and these expressions are developed in the paper.

Aidarous, in [9], proposes a method for locating a fixed number of noiseless actuators in a continuous stochastic DPM such that a functional of type (2.3b) is minimized. The technique involves approximating the general control functions (i.e. $u(t)$ in (2.1)) by a linear combination of a finite number of coordinate functions. The cost functional then becomes a function of the coefficients in this linear expansion of $u(t)$ and the spatial coordinates of the actuator locations. The coefficients and spatial coordinates which minimize the cost functional are then identified by applying a gradient-type algorithm. The algorithm uses a finite approximation of the system's Green function (very similar to the impulse function for an LPM) and first order variations of the cost functional with respect to the coefficients and spatial coordinates in order to develop gradient-type update equations for the coefficients and spatial coordinates. In [10] Aidarous et al. present the discrete time version of the algorithm just discussed.

The question of locating a fixed number of noisy sensors and noiseless actuators so that the controller designed to minimize a functional of the form (2.3b) achieves the smallest possible value of (2.3b) is addressed by Ichikawa and Ryan in [11]. Their technique for finding the optimal locations is to adopt a finite dimensional eigenfunction expansion for the operator A_s in (2.1) and then plot the value of the cost functional for all admissible sensor and actuator

configurations. The paper presented an SAS example for the stochastic diffusion equation where measurement noise was assumed but process noise was not (i.e. $v(t,s) \neq 0$, $w(t,s) = 0$ in (2.1)). The problem was to select 1 out of 16 possible actuator locations and 1 out of 16 possible sensor locations (i.e. $\bar{m} = \bar{\ell} = 1$, $m = \ell = 16$). The interesting result from the example was that different optimal locations occurred when the sensor and actuator problem was solved simultaneously as opposed to separately and the lowest cost value was associated with the simultaneous solution.

Juang and Rodriguez, in [12], further demonstrated the effect that actuator location can have on the performance of a closed-loop regulator designed to minimize a quadratic cost functional of type (2.3). The chosen model was an LPM for a simply supported beam with no process noise (i.e. $D = 0$ in (1.1)). One actuator along with one noisy sensor were used in the regulator design. As in the case of [11], the optimal actuator location was determined by plotting the value of (2.3a) versus the admissible actuator locations. The results showed that the optimal actuator location was a function of both model complexity (i.e. the dimension of the LPM used to approximate the DPM), and the weights chosen for the quadratic performance index.

The theory of Input Cost Analysis (ICA) and Output Cost Analysis (OCA) was used in references [13]-[16] to pose algorithms for the selection of \bar{m} and of m actuators and $\bar{\ell}$ out of ℓ sensors to minimize cost functionals of type (2.3a). The fundamental concept behind (ICA) and (OCA) is to determine the contribution that each admissible input and output is making to the cost functional, and then use this information

to make sensor and actuator selection decisions. The specific details of ICA and OCA are fundamental to this research and are presented in Chapter 5. In [13] and [14] Skelton and Chiu lay the ground work for ICA and OCA. Analytic expressions are developed in [13] (for $u(t) = 0$) which specify the contributions that the columns of $D(d_1, d_2, \dots, d_p)$ and columns of $C^T(c_1, \dots, c_k)$ are making to the cost functional. These expressions have been labeled 'parametric' ICA and OCA since they involve the coefficients associated with the inputs and outputs. In [14] analytic expressions are developed (for $u(t) = 0$) which specify the contributions that the individual inputs ($w_i, i = 1, \dots, p$) and outputs ($y_i, i = 1, \dots, k$) are making to the cost functional. The application of these results to the more meaningful closed-loop (i.e. $u(t) \neq 0$) SAS questions was achieved by Chiu, Skelton, and DeLorenzo in [15] and [16].

Under the assumption that the actuators have no noise, Chiu in [16] shows that increasing the number of actuators will never increase the cost functional. [Theorem 1, 15] He also states that increasing the number of sensors will never increase the cost functional [Theorem 2, 15]. Given these results, Chiu develops an SAS algorithm which suggests a specific number of sensors and actuators as well as specifies their desired locations. The algorithm uses closed-loop 'parameteric' versions of ICA and OCA to determine a suggested number of actuators and sensors and then an iterative search routine is used to determine the sensor and actuator locations which satisfy the necessary conditions for minimization of the cost functional as derived from the matrix minimum principle of Athans.[46] The iterative search routine is an 'extended' version of the search routine proposed by Chen and Seinfeld which will be discussed shortly.

In [16], Skelton and DeLorenzo developed closed-loop ($u(t) \neq 0$) analytical expressions for the contribution that each actuator (u_i), process noise source (w_i), sensor noise source (v_i) and output (y_i) makes to the cost functional. These expressions were combined to form the actuator and sensor effectiveness values mentioned in the introduction and discussed in Chapter 5. The SAS algorithm based on these effectiveness values will also be discussed in detail in Chapter 5 and is not be repeated here. The main differences between [16] and the work of Chiu in [15] are that noisy actuators are considered in [16] which invalidates [Theorem 1, 15], the 'parametric' version of ICA and OCA which is applied to the closed-loop situation in [15] produces calculations which are mathematically complex and computationally burdensome compared to the non-parametric calculations of [16], and the iterative search routine of [15] involves calculations beyond the closed-loop 'parametric' ICA and OCA calculations while the search routine in [16] involves no calculations beyond the closed-loop ICA and OCA calculations.

2.3.2 SAS Based the Error Covariance Matrix

References [17]-[24] are concerned with locating a fixed number of sensors in an LPM or DPM so that either (2.2a) or (2.2b) is minimized over all possible configurations. In [17] Yu and Seinfeld discuss this problem for a DPM whose state vector $x(t,s)$ can be represented by a finite number of eigenfunctions of A_s . The system then becomes essentially an LPM and it is possible to develop an ordinary matrix differential equation for the error covariance matrix P in terms of the spatial coordinate s . Yu and Seinfeld then propose a sub-optimal algorithm which sequentially locates sensors so that the trace of the steady state

P matrix is minimized subject to the constraint that only the location of the next sensor to be added may be varied: the previously located sensors being assumed fixed. The optimal location, for the next sensor is determined from the integration of the ordinary differential equation for P in terms of s . The sub-optimality of this algorithm was verified by the work of Colantuoni and Padmanabhan in [25].

Chen and Seinfeld develop an algorithm in [18] which locates $\bar{\lambda}$ out of λ sensors to minimize criterion (2.2.b). The algorithm is, iterative and searches for the $\bar{\lambda}$ sensor locations that satisfy the necessary conditions for minimizing (2.2b) as derived from a distributed parameter formulation of the matrix minimum principle. The search routine requires no gradient calculations, but does require the solution of a partial differential Riccati equation at each iteration and the calculation of a switching function for all λ sensor locations. The switching functions are based on the spatial integration of a functional which is quadratic in the P operator specified by the current set of $\bar{\lambda}$ sensors.

In [19] Aidarous et. al. propose using a finite coordinate function expansion for P and the DPM Green's function along with a modified conjugate gradient algorithm to develop update equations for the \bar{m} sensor locations which minimize (2.2b). The algorithm is essentially the dual of the actuator selection algorithms of [9] and [10]. The necessary conditions for the convergence of the algorithm in [19] is presented in [20].

Kumar and Seinfeld in [21] propose choosing sensors to minimize the trace of an upperbound of P where the calculation of this

upper bound does not involve the solution of a partial differential Riccati equation. However, it does involve an orthonormal approximation to the systems Green's function. A gradient type algorithm based upon this upper bound expression is proposed to update the sensor locations. An example which placed two sensors in a one dimensional heat conduction equation is presented, and the results are compared with the algorithm of [18]. The answers compared favorably, and a significant savings in computation resulted when the upper bound criterion was used in place of P .

Omatu et al. in [22] recommend adopting the following approximation for P :

$$(2.4) \quad P(t, s_1, s_2) \approx \sum_{i,j=1}^N P_{ij}(t) \phi_i(s_1) \phi_j(s_2); \quad N \ll \infty$$

where the functions ϕ_i, ϕ_j are eigenfunctions for A_s . Using this approximation and the comparison and existence results for Riccati equations derived in the paper, a set of necessary and sufficient conditions are developed for locating $\bar{\ell}$ out of ℓ sensors in a DPM of type (2.1), (2.2) so that $\text{tr}\{P(t)\}$ is minimized. These conditions essentially involve definiteness comparisons of the matrix product:

$$(2.5) \quad \sum_{i,j=1}^{\bar{\ell}} \phi^T M^T(t, s_i) V^{-1}(t, s_i, s_j) M(t, s_j) \phi$$

for each admissible set of $\bar{\ell}$ sensor coordinates, s . The matrix ϕ is an $N \times \bar{\ell}$ matrix of eigenfunctions evaluated at the $\bar{\ell}$ spatial locations. The necessary and sufficient conditions based on (2.5) do not require the

calculation of a partial differential Riccati equation or complex gradients but, as posed, do require a direct search of all admissible sensor configurations.

In [23] Wei and Wu address the problem of locating \bar{l} out of l sensors for an LPM of type (1.1) to minimize criterion (2.2a). The matrix maximum principle is used to derive necessary conditions for the sensor locations. A sufficient condition which involves the definite comparison

$$(2.6) \quad P_k M_k^T V_k^{-1} M_k P_k \geq P_k M_j^T V_j^{-1} M_j P_k$$

is then derived where the subscripts k, j represent admissible configurations of \bar{l} sensors. If (2.6) holds for some k over all possible configurations then k is the optimal configuration. A direct search algorithm is proposed to find k . The results of this paper are in many respects the LPM version of the results of [22]. Finally, the paper also suggested minimizing the trace of an upper bound on P instead of (2.2a) in order to avoid solving a Riccati equation at iteration.

A technique for adjusting the elements of the measurement matrix M in (1.1) was suggested by Arbel in [24]. The goal of the technique is to adjust the elements of M so that a weighted trace of the error covariance P is minimized subject to location constraints on the elements of M . The first order variation of P to each element M is calculated via a Lyapunov type equation and these variations are used to adjust the elements of the M matrix as long as the constraints on M are not violated.

2.3.3 Other SAS Criterion

References [26]-[43] pose SAS questions for criterion other than the minimization of (2.2) or (2.3). In many cases the criteria are minor modifications of (2.2) or (2.3), while the criteria chosen by some references ([31]-[35]) do not apply to the context of this research. These references are included, however, for completeness.

In [26], Curtain and Ichikawa propose the selection of sensors to minimize both the cost of taking a measurement and the trace of the covariance of the estimation error for distributed systems. No specific form for the cost of taking a measurement is offered, and a direct search technique is used for those distributed systems whose solutions may be expressed in terms of eigenfunctions of A_s . Amouroux et. al. in [27] add a term which includes certain variables in the control law to the criterion (2.2b). Given a fixed number of sensors in a linear stochastic DPM, these sensors are then located to minimize the cost functional of the chosen control law variables and the error covariance integral. A modified gradient algorithm is suggested to update the sensor locations.

In [28] Morari and O'Dowd pose a sensor selection problem for a DPM driven by non-stationary noise. It is shown that, in general, the presence of non-stationary noise makes the system unobservable. A projection technique is derived which yields an observable system and a Kalman-Bucy filter for the observable system is constructed. The sensors are then selected to minimize the error caused by the unobservable non-stationary noise components. The criterion which accomplishes this is shown to be the trace of spatial integral of P for

the steady state Kalman filter of the projected system. The criterion is applied by using a direct search of the sensor locations.

Kosut et. al. discuss an SAS reliability question for a general LPM with a fixed set of sensor locations in [30]. The "reliability" question is approached by introducing a parameter representing sensor systematic error (i.e. axes misalignment, scale factor and bias error, etc.) into the measurement equation. An optimization problem is formulated which places the sensors, with varying degrees of systematic error, in the fixed locations such that a lower bound on $P(t)$ is minimized. Necessary conditions for the optimization are derived, but no technique of solution is offered.

The problem of selecting at each instant of time, one measurement provided by one out of many sensors in a linear stochastic LPM is addressed by Athans in [31]. The criterion for selection is the minimization of a weighted combination of $P(t)$ and a term reflecting observation cost. The observation cost term is expressed by functions which denote the per-unit-of-time cost of making an observation. The problem is transformed into a deterministic optimal control problem and the matrix maximum principle is used to derive the necessary conditions for optimality. A specialized gradient algorithm is used in obtaining the solution. In [32], Herring and Melsa extend the results of Athans [31] to the selection of a best combination of measurement devices instead of selection of a best single device. A similar situation is addressed in [33] where a measurement subsystem for a discrete time linear decentralized LPM is chosen from a number of subsystems. An "ideal" subsystem which minimizes the trace of the steady state P

matrix is found and then the existing subsystem which is closest to this ideal subsystem, as determined by a certain information measure, is chosen.

In [34] and [35] the problem of selecting at each instant of time one out of many actuators or sets of actuators to achieve a desired result is addressed. Vanbeveren and Gevers, in [34], consider a discrete deterministic linear LPM and the minimization of a criterion (2.3a). Propositions which specify when certain possible sequences of actuator choices can be ignored are proved and these propositions are combined with a proposed "decision tree" method to arrive at a solution. The computational burden of this method for large systems is noted and a suboptimal "forward-backward" algorithm is proposed for this case. In [35] Martain uses the theory of relaxed controls to solve the problem posed in [34] for both an LPM and DPM. For the LPM, the existence of a solution is established and necessary conditions for optimality are derived. A steepest decent gradient algorithm is proposed to find the solution to the necessary conditions.

In [36], Mehra discusses placing a fixed number of sensors in a linear stochastic OPM. The problem is to select the sensor locations that minimize a norm of the inverse of the Fisher information matrix subject to a norm constraint on the measurement matrix (M). The necessary conditions for optimality are derived through the lagrange multiplier technique. However, a procedure for determining an appropriate measurement norm constraint is not given.

Placing a fixed number of sensors in a matrix second order LPM which is written in modal form (i.e., mass matrix is identity, stiffness

matrix is diagonal) is considered by Buhariwala in [37]. The minimization criterion is a weighted sum of each mode's observability norm, and the sensor locations that minimize this sum are solved for by a psuedo-random search algorithm.

In [37] and [39] the problem of locating sensors in a DPM is again considered. Ewing and Higgins, in [38], recast the partial differential equation for the system in the form of a variational functional. The sensors are then placed in optimal locations by choosing the set of locations that minimizes the derived variational functional. A steepest decent gradient algorithm is used to find the minimizing sensor locations, and if only discrete locations are available for the sensors, a location constraint must be developed. In [39] Caravani and Phillo approximate the DPM by a finite eigenfunction expansion. They then propose finding the sensor locations that minimize the expected value of the square estimation error when the residual components of the true initial state are allowed to vary within a sphere of given radius. An analytical expression for the above defined criterion is derived and a direct search algorithm is used in a simple example having one sensor.

The problem of locating actuators (noiseless) in linear deterministic oscillatory systems is considered by Arbel and Gupta in [40] and Arbel in [41]. The control objective is assumed to be open-loop minimum-energy control and the minimum energy value is shown to depend directly on the controllability matrix. An actuator selection algorithm is thus proposed based upon the minimization of a measure of the controllability matrix. The gradient type algorithm requires the system be placed in Jordan canonical form and makes use of the diagonal dominance

property of the controllability matrix for systems in canonical form [Theorem 1, 40].

In [42] VanderVelde and Carigan define a measure of controllability which provides a quantitative indication of how well a *deterministic* linear LPM can be controlled with a given set of actuators. The effect of component unreliability is introduced by computing the expected value of this controllability measure accounting for the likelihood of various combinations of component failures. A direct search algorithm is then used to locate the actuators of the system to minimize this "controllability/reliability" criterion. The process of defining this criterion for a large number of actuators and then applying a direct search or integer programming (for discrete locations) type algorithm is, as noted by the authors, computationally burdensome.

Lindberg and Longman in [43] discuss actuator placement for a linear *deterministic* LPM. They use the concept of modal space control developed by Meirovitch et. al. but eliminate the need for a large number of actuators (i.e. one for each system mode) by introducing a psuedo-inverse for the control distribution matrix (B^+). The main advantage of modal space control, (i.e. the elimination of the need to solve large order Riccati equations) is retained. The goal is to locate the actuators in the system to achieve minimum energy control, and this is achieved by finding the set of actuators which minimizes the largest singular value of B^+ . Either a direct or gradient search technique is proposed to accomplish the minimization.

2.4 General Relation to the SAS Problem

As mentioned in the beginning of the Chapter, none of the criterion used in the reviewed SAS literature directly addresses the specific steady state variance constraints (σ^2, μ^2) defined in (1.8). The criteria of (2.2) and (2.3) and the other related SAS criteria all have merit, and from an intuitive stand point, should indirectly attempt to satisfy (σ^2, μ^2) . However, an SAS criterion which directly encompasses (σ^2, μ^2) would be more desirable, and the LQG weight selection algorithm in Chapter 4 provides one method for achieving such a goal.

With the exception of [16], none of the literature considered the selection of noisy actuators (i.e. B also becomes a partition of D). The results of [16] are discussed extensively in Chapter 5, but have shown that [Theorem 1, 15] is not valid when noisy actuators are considered. Therefore, noisy actuators can significantly change the complexion of actuator selection and merit consideration.

Most of the literature used some form of gradient or direct search technique to determine the sensors and actuators which satisfied the necessary and/or sufficient conditions derived from the particular selection criterion considered ([15], [16] and [18] are notable exceptions). As discussed in the introduction, even for a modest number of sensors and actuators a direct search of all possible sensor and actuator locations becomes infeasible. Furthermore, when sensor and actuator selection is necessary for large complex systems with many admissible sensor and actuator locations the requirement for any form of gradient calculation is severe and can easily become prohibitive, particularly if spatial integrations are also required. A search

algorithm which does not require direct search or gradient calculations would certainly be desirable in the above situations and only algorithms with these properties were considered in this research. The detailed discussion of the approach this research took to the SAS problem begins with a discussion of LQG theory.

3.0 LQG THEORY

As noted in the Introduction, an important step in the solution of the SAS problem is the specification of a control law, and LQG theory has been chosen to specify the control law in this research. The theory originated in the 1960's with the foundational work of Kalman, Bucy and others. Since that time, many papers and texts have been written which further clarify and expand the theory. References [1]-[3] are a few examples. Section 3.1 contains the fundamental results of LQG theory when applied to systems of type $S(n,k,m,\ell)$ (i.e. (1.1)-(1.3)) and long periods of control are required ($t \rightarrow \infty$). The proofs of the results are omitted but are readily available in the previously mentioned references. In section 3.2 the LQG version of the SAS problem is formulated and section 3.3 discusses the advantages and disadvantages of this formulation.

3.1 The Steady State LQG Controller

For a system of type $S(n,k,m,\ell)$ LQG theory guarantees a stable closed-loop system and a closed-loop dynamical controller ($u(z(\tau))$, $0 < \tau \leq t$) which minimizes the following cost functional:

$$(3.1) \quad V_0 = \lim_{(t_1 - t_0) \rightarrow \infty} \frac{1}{(t_1 - t_0)} E \int_{t_0}^{t_1} \{y^T(t)Qy(t) + u^T(t)Ru(t)\}dt; Q, R > 0$$

The defining equations for this steady-state controller are as follows:

$$(3.2a) \quad u(t) = -R^{-1}B^TK\hat{x}(t) \triangleq G\hat{x}(t)$$

$$(3.2b) \quad \dot{\hat{x}}(t) = A_c\hat{x}(t) + Fz(t) ; \quad F \triangleq PM^TV^{-1} ; \quad A_c \triangleq A + BG - FM$$

$$\hat{x}(t_0) = x_0$$

$$(3.2c) \quad KA + A^TK - KBR^{-1}B^TK + C^TQC = 0 ; \quad (\text{Control Riccati Equation})$$

$$(i.e. \lim_{t \rightarrow \infty} K(t) = K)$$

$$(3.2d) \quad PA + AP - PM^TV^{-1}MP + DWD^T = 0 ; \quad (\text{Filter Riccati Equation})$$

$$(i.e. \lim_{t \rightarrow \infty} P(t) = P)$$

Substituting this controller into $S(n,k,m,l)$, the following $2n$ dimensional closed-loop system results:

$$(3.3a) \quad \dot{x} = Ax + Bw ; \quad x \triangleq (x^T \hat{x}^T)^T \quad w \triangleq (w^T v^T)^T$$

$$y = Cx ; \quad y \triangleq (y^T, u^T)^T$$

where,

$$(3.3b) \quad A \triangleq \begin{bmatrix} A & BG \\ FM & A_c \end{bmatrix} ; \quad B \triangleq \begin{bmatrix} D & 0 \\ 0 & F \end{bmatrix} ; \quad C \triangleq \begin{bmatrix} C & 0 \\ 0 & G \end{bmatrix}$$

and

$$(3.3c) \quad v_0 \triangleq \lim_{(t_1-t_0) \rightarrow \infty} \frac{1}{(t_1-t_0)} E \int_{t_0}^{t_1} y^T(t) Q y(t) dt ; \quad Q = \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix}$$

Given the noise characteristics of $S(n,k,m,l)$, the steady-state variance matrix for $S(2n, k+m, m+l, 0)$ is known to be:

$$(3.4) \quad E_{\infty}\{(x - 0)(x - 0)^T\} = E_{\infty}\{x x^T\} = \begin{bmatrix} \hat{X} + P & \hat{X} \\ \hat{X} & \hat{X} \end{bmatrix}$$

where,

$$(3.5) \quad \hat{X} = \lim_{t \rightarrow \infty} \hat{X}(t) = E_{\infty}\{(\hat{x} - 0)(\hat{x} - 0)^T\} = E_{\infty}\{\hat{x} \hat{x}^T\},$$

and is the solution of the following steady-state Lyapunov equation:

$$(3.6) \quad \hat{X}(A + BG)^T + (A + BG)\hat{X} + PM^T V^{-1} MP = 0.$$

Also, coupling the definition of x in (3.3a) with (3.4) gives,

$$(3.7) \quad X = \lim_{t \rightarrow \infty} X(t) = E_{\infty}\{x x^T\} = \hat{X} + P$$

Assuming that the weighting matrices of V_0 in (3.1) are strictly diagonal, and using the results of (3.2)-(3.6) along with some linear algebra, the following sequence of manipulations of V_0 produces a useful result. First, using the fact that $x^T A x = \text{tr}\{x x^T A\}$, V_0 becomes:

$$(3.8) \quad V_0 = \lim_{(t_1 - t_0) \rightarrow \infty} \frac{1}{(t_1 - t_0)} \text{tr} \int_{t_0}^{t_1} [CX(t)C^T Q + R^{-1}B^T K(t)\hat{X}(t)K(t)B] dt$$

Since $(t_1 - t_0) \rightarrow \infty$, the value of V_0 will be dominated by the steady state portion of the integral in (3.8) and the LQG controller must, therefore, minimize the following expression also: [1]

$$(3.9) \quad V \triangleq \lim_{(t_1 - t_0) \rightarrow \infty} \frac{1}{(t_1 - t_0)} \text{tr} \int_{t_0}^{t_1} [CXC^T Q + R^{-1}B^T K\hat{X}KB] dt$$

The integrand of (3.9) is now time-invariant and this means (3.9) can be written as:

$$(3.10) \quad \nu = \text{tr}[CXC^TQ + R^{-1}B^TKXB] ,$$

which is directly equivalent to the following:

$$(3.11a) \quad \nu = \text{tr}[E_{\infty}\{yy^T\}Q + E_{\infty}\{uu^T\}R] ,$$

or the more well known form

$$(3.11b) \quad \nu = E_{\infty}[y^TQy + u^TRu] .$$

Invoking the assumed diagonality of Q and R yields the following useful expression which the LQG controller is *known to minimize*.

$$(3.12) \quad \nu = \sum_{i=1}^k E_{\infty}y_i^2q_i + \sum_{i=1}^m E_{\infty}u_i^2r_i ,$$

where q_i and r_i represent the i^{th} diagonal entries of Q and R respectively.

3.2 LQG Theory and the SAS Problem

The two key quantities in the SAS problem are $E_{\infty}y_i^2$ and $E_{\infty}u_i^2$. As might be guessed from the results of the previous section, analytic expressions for these quantities are readily obtained when an LQG controller is used for the system $S(n,k,m,\ell)$. The expression for $E_{\infty}y_i^2$ may be derived as follows. By definition, $y_i(t)$ can be written as:

$$(3.13) \quad y_i(t) = c_i^T x(t)$$

where c_i^T is the i^{th} row of C . Therefore, $y_i^2(t)$ becomes

$$(3.14) \quad y_i^2(t) = (c_i^T x(t))(c_i^T x(t)) ,$$

or since the transpose of a scalar is still the scalar:

$$(3.15) \quad y_i^2(t) = c_i^T x(t) x^T(t) c_i$$

Letting $t \rightarrow \infty$, taking the expectation of (3.15) and making use of (3.7), the desired expression for $E_{\infty} y_i^2$ results:

$$(3.16) \quad E_{\infty} y_i^2 = c_i^T (\hat{X} + P) c_i .$$

The derivation for $E_{\infty} u_i^2$ proceeds in a similar fashion by first using (3.2a) to develop this expression for u_i :

$$(3.17) \quad u_i(t) = -r_i^{-1} b_i^T K \hat{x}(t)$$

where b_i is the i^{th} column of B . Then, $u_i^2(t)$ can be written as

$$(3.18) \quad u_i^2(t) = r_i^{-2} b_i^T K \hat{x}(t) \hat{x}^T(t) K b_i ,$$

where the fact that $K = K^T$ (i.e. matrix Riccati solutions are symmetric) has been used. Letting $t \rightarrow \infty$ and taking the expectation of (3.18) results in the following analytic expression for $E_{\infty} u_i^2$:

$$(3.19) \quad E_{\infty} u_i^2 = r_i^{-2} b_i^T \hat{K} \hat{X} K b_i \quad .$$

It should be noted that (3.16) and (3.19) are implicit functions of the weighting matrices Q and R and that (3.16) is an explicit function of r_i . Therefore the values of $E_{\infty} y_i^2$ and $E_{\infty} u_i^2$ may be changed by adjusting the components of Q and R . In fact, the following general trends are known. Increasing (decreasing) r_i will decrease (increase) $E_{\infty} u_i^2$ while causing a general increase (decrease) in the remaining $E_{\infty} u_j$, $j \neq i$ and a general increase (decrease) in $E_{\infty} y_i^2$, $i = 1, \dots, k$. Similarly increasing (decreasing) q_i will decrease (increase) $E_{\infty} y_i^2$ while causing a general decrease (increase) in the remaining $E_{\infty} y_j$, $j \neq i$ and a general increase (decrease) in $E_{\infty} u_i^2$, $i = 1, \dots, m$.

Incorporating the LQG theory of (3.2)-(3.7) and the expressions (3.16) and (3.19) into the SAS problem defined in the introduction produces the following parameter optimization or non-linear programming problem which will be referred to as the SASLQG problem.

SASLQG Problem Statement

Given: A system of type $S(n, k, m, \ell)$ which has only \bar{m} out of m actuators and $\bar{\ell}$ out of ℓ sensors available for the design of a steady state LQG regulator which must achieve (σ^2, μ^2) .

Required: Specify the closed-loop system which satisfies the following input-constrained or output-constrained requirements:

Input-constrained

If (σ^2, μ^2) are achievable,

$$(3.20a) \quad \min_S \sum_{i=1}^k E_{\omega} y_i^2 / \sigma_i^2 \quad \text{subject to} \quad E_{\omega} u_i^2 = \bar{\mu}_i^2 \quad \forall i = 1, \dots, \bar{m}$$

else,

$$(3.20b) \quad \min_S \sum_{i=1}^k E_{\omega} y_i^2 / \sigma_i^2 \quad \forall i: E_{\omega} y_i^2 > \sigma_i^2$$

$$\text{Subject to } E_{\omega} u_i^2 = \bar{\mu}_i^2 \quad \forall i = 1, \dots, \bar{m}$$

Output-constrained

If (σ^2, μ^2) are achievable,

$$(3.21a) \quad \min_S \sum_{i=1}^{\bar{m}} E_{\omega} u_i^2 / \bar{\mu}_i^2 \quad \text{subject to} \quad E_{\omega} y_i^2 = \sigma_i^2 \quad \forall i = 1, \dots, k$$

else,

$$(3.21b) \quad \min_S \sum_{i=1}^{\bar{m}} E_{\omega} u_i^2 / \bar{\mu}_i^2 \quad \forall i: E_{\omega} u_i^2 > \bar{\mu}_i^2$$

$$\text{Subject to } E_{\omega} y_i^2 = \sigma_i^2 \quad \forall i = 1, \dots, k$$

The essential difference between the SAS problem and the SASLQG problem is that the SAS problem is requiring the best choice of sensors and actuators and controller structure to achieve (σ^2, μ^2) , while the SASLQG problem assumes an LQG controller and looks for the best choice of sensors and actuators to achieve (σ^2, μ^2) . There is *no* guarantee that the SASLQG problem is *the* solution to the SAS problem.

3.3 LQG Theory Advantages and Disadvantages

In light of the discussion of Section 3.2, the advantages and disadvantages of the LQG approach to the SAS problem can be summarized.

Advantages of the LQG Approach to SAS

- (1) LQG theory provides a closed-loop linear dynamical controller and necessary and sufficient conditions for closed-loop asymptotic stability (i.e. (1.3))
- (2) The controller computations of (3.2) and (3.6) are straight forward (non-iterative) and the constant gain matrices are easy to implement.
- (3) Analytical expressions exist for $E_{\omega u_i}^2$ and $E_{\omega y_i}^2$ and require no additional major calculations beyond those for the controller.
- (4) The values of $E_{\omega u_i}^2$ and $E_{\omega y_i}^2$ can be changed by adjusting the components of Q and R and those Q and R adjustments do *not* effect the filter Riccati solution P (3.2d) (i.e. the separation theorem applies).

Disadvantages of the LQG Approach to SAS

- (1) The method for adjusting the elements of Q and R to achieve desired changes in $E_{\omega u_i}^2$ and $E_{\omega y_i}^2$ has, in the past, been trial and error.
- (2) Solution of the SASLQG problem does not guarantee that the SAS problem has been solved. (i.e. another control structure might do better)
- (3) The order of the controller is usually the order of the model (for exceptions see [64]).

Disadvantage (1) has been eliminated by the Q and R selection procedure developed in this research and presented in Chapter 6. Also, the sensor and actuator effectiveness values derived in Chapter 5 are fundamentally rooted in LQG theory and provide another major incentive for using LQG theory in the SAS problem. Concerning disadvantages (2) and (3), they have been treated as 'current' necessary evils for gaining insight into the SAS problem. Before continuing with the development of the actuator and sensor effectiveness values and the weight selection algorithm, Chapter 4 will discuss the physical systems chosen for practical application of this research and define their models.

4.0 THE LARGE SPACE STRUCTURE (LSS)

The design algorithm developed by this research for the SASLQG problem has been tested on two substantial LSS models. The results are presented in Chapter 7, while the details of the models are presented in this chapter. Section 4.1 contains a general description of an LSS and its typical mission and control requirements. The general DPM for an LSS and an outline of a technique which generates a matrix second order ordinary differential equation from the DPM is presented in Section 4.2. In Section 4.3 a model of type $S(n,k,m,\ell)$ is developed for the hoop column antenna satellite from finite element NASTRAN data, and a specific SAS problem for the hoop antenna is then posed. The $S(n,k,m,\ell)$ model for a solar optical telescope is presented in Section 4.4 and an SAS problem is also posed for this model.

4.1 General LSS Description and Mission

The recent successes of the Space Shuttle has made the large space structure (LSS) an imminent reality. These future space structures will be measured in kilometers and, of necessity, will be lightweight and highly flexible (light damping). Standard LSS missions will include power generation, surveillance, astronomy, and communications. These missions will require stringent pointing accuracy, shape control and vibration suppression. To satisfy the demanding mission requirements the LSS may require an active, regulator-type controller

with multiple sensors and actuators located throughout the structure [67] - [72]. Furthermore, given the size of an LSS, there will be a large set of admissible sensor and actuator locations. The controls engineer is then faced with the problem of determining where to locate a limited number of sensors and actuators to 'best' achieve the LSS mission (i.e. he must solve the SAS problem!) Therefore, the problem of regulating an (LSS) presents an excellent proving ground for SAS techniques. For this reason two LSS models have been chosen to test the SAS design algorithm proposed by this research. Their descriptions follow.

4.2 The LSS Model

The problem of modeling an LSS is well discussed in the literature. A representative example is [67]. The modeling process centers around discretizing a system of partial differential equations of the following form:

$$(4.1) \quad \begin{cases} m(s) \frac{\partial^2 \Omega(t,s)}{\partial t^2} + D_s \frac{\partial \Omega(t,s)}{\partial t} + A_s \Omega(t,s) = \sum_{i=1}^m \delta(s-s_i^a) f_i u_i(t) \\ y(t,s) = c_p \Omega(t,s) + c_r \frac{\partial \Omega(t,s)}{\partial t} ; \quad y, c_p, c_r \in R^k \text{ (system outputs)} \\ z(t)_j = m_{p_j} \Omega(t, s_j^s) + m_{r_j} \frac{\partial \Omega(t, s_j^s)}{\partial t} ; \quad j = 1, \dots, l \\ \text{(system measurements)} \end{cases}$$

where, as in (2.1), s represents the spatial coordinates defined in the domain S . The quantity $\Omega(t,s)$ represents the translational and rotational motions for each point in S and could be a vector but for

notational simplicity it will be assumed scalar. The expression $\sum_{i=1}^m \delta(s-s_i^a) f_i u_i(t)$, where f_i is a scalar influence coefficient, is used to represent the forcing functions of m actuators located at points s_i^a , and $m(s)$ is the mass density which is positive and bounded on S . A_s is a time-invariant symmetric, non-negative differential operator and is assumed to have a discrete spectrum defined by:

$$(4.2a) \quad A_s \phi_i(s) = \lambda_i \phi_i(s) ; \quad i = 1, 2, \dots \infty$$

$$A_s^{1/2} \phi_i(s) = \lambda_i^{1/2} \phi_i(s) ; \quad i = 1, 2, \dots \infty$$

and the eigenfunctions $\phi_i(s)$ in (4.2) are assumed to be orthogonal with respect to $m(s)$ which implies the following:

$$(4.2b) \quad \int_S m(s) \phi_i(s) \phi_j(s) ds = \begin{cases} m_j & \text{when } i = j \\ 0 & \text{otherwise} \end{cases}$$

The operator D_s generates the damping term for the structure and is currently not well understood. It usually consists of a skew symmetric part which represents gyroscopic damping due to on-board rotors or a constant spin of the LSS, and a small symmetric part which represents the internal structural damping [68]. It is this symmetric part of D_s which is hardest to understand, and for mathematical convenience it is usually modeled as:

$$(4.3) \quad D_s (\text{symmetric}) = 2\zeta A_s^{1/2} \quad (\text{i.e. similar to } 2\zeta w \text{ for an LPM})$$

The symbol ζ is taken to be a matrix of damping ratios the components of which are small (i.e. around .005). The vector $y(t,s)$ in (4.1) represents the outputs of the system which are to be regulated. They are represented by linear combinations of position ($c_p \Omega(t,s)$) and rate ($c_r \frac{\partial \Omega(t,s)}{\partial t}$), where c_p and c_r are assumed to be constant $k \times 1$ vectors. The sensor information $z(t)$ for (4.11) is assumed to be provided by a set of ℓ sensors located at points s_j^S which measure both position and rate. For the i^{th} position measurement, the rate influence coefficient m_{r_i} will be zero, and for the i^{th} rate measurement m_{p_i} will be zero. It should also be noted, that for the moment, actuator and sensor noise, along with model uncertainty have been neglected.

If the \mathcal{D}_S operator is assumed to have no gyroscopic terms, and (4.3) is used to represent \mathcal{D}_S the instantaneous position of the LSS can be represented by an infinite sum of the orthogonal eigenfunctions of A_S with strictly time varying coefficients.

$$(4.4) \quad \Omega(t,s) = \sum_{i=1}^{\infty} q_i(t) \phi_i(s) ;$$

Substituting (4.4) into (4.1) and successively multiplying by $\phi_j(s)$ and integrating over S , the LSS may be represented by the following infinite set of ordinary differential equations when the orthogonality of $(m(s) \phi_i(s) \phi_j(s))$ along with the spectrum definitions for A_S and $A_S^{1/2}$ are used.

$$m_j \ddot{q}_j(t) + 2\zeta_j \lambda_j^{1/2} \dot{q}_j(t) + \lambda_j q_j(t) = \sum_{i=1}^m \phi_j(s_i^a) f_i u_i(t); j=1, \dots, \infty$$

$$(4.5) \quad y(t,s) = c_p \sum_{i=1}^{\infty} \phi_i(s) q_i(t) + c_r \sum_{i=1}^{\infty} \phi_i(s) \dot{q}_i(t);$$

$$z_j(t) = m_{p_j} \sum_{i=1}^{\infty} \phi_i(s_j^s) q_i(t) + m_{r_j} \sum_{i=1}^{\infty} \phi_i(s_j^s) \dot{q}_i(t); j = 1, \dots, \ell$$

Three points of interest arise in (4.5). The first being that the output $y(t,s)$ is still a vector of both time and space and usually represents such things as line-of-sight (LOS) angles, defocus lengths, shape information etc. For practical reasons, usually from a control design standpoint, it is necessary to discretize $y(t,s)$ in a manner similar to $z(t)$. i.e.

$$(4.6) \quad y_j(t) = c_{p_j} \sum_{i=1}^{\infty} \phi_i(s_j^y) q_i(t) + c_{r_j} \sum_{i=1}^{\infty} \phi_i(s_j^y) \dot{q}_i(t); j=1, \dots, k$$

The quantities c_{p_j} and c_{r_j} represent the j^{th} element of c_p and c_r respectively.

The second point of interest in (4.5) is that infinite sums are involved which again, from a practical stand point, cannot be handled. To alleviate this problem, some form of the model analysis can be performed to choose a finite number (N) of eigenfunctions and eigenvalues (i.e. modes) to represent the LSS. With this modal truncation, (4.5) and (4.6) can be written as follows:

$$(4.7) \quad \begin{cases} M\ddot{\tilde{q}}(t) + D\dot{\tilde{q}}(t) + K\tilde{q}(t) = Fu(t) ; \quad \tilde{q} \in \mathbb{R}^N \\ y(t) = [C_p \quad C_R] \begin{bmatrix} \tilde{q}(t) \\ \dot{\tilde{q}}(t) \end{bmatrix} \\ z(t) = [M_p \quad M_R] \begin{bmatrix} \tilde{q}(t) \\ \dot{\tilde{q}}(t) \end{bmatrix} \end{cases}$$

Where M is an $N \times N$ diagonal matrix with m_j , $j = 1, \dots, N$ on the diagonal. The elements of $q(t)$ are the N $q_i(t)$'s associated with the N chosen modes and the matrices D and K are $N \times N$ diagonal matrices containing the following data for the selected modes:

$$(4.8) \quad D = \text{diag} [2 \zeta_1 \lambda_1^{1/2} \dots 2 \zeta_N \lambda_N^{1/2}]$$

$$(4.9) \quad K = \text{diag} [\lambda_1 \dots \lambda_N]$$

The matrix F is an $N \times m$ matrix whose elements consist of eigenfunctions evaluated at the actuator locations.

$$(4.10) \quad F = \begin{bmatrix} \phi_1(s_1^a)f_1 & \phi_1(s_2^a)f_2 & \dots & \phi_1(s_m^a)f_m \\ \vdots & & & \\ \phi_N(s_1^a)f_1 & \dots & \dots & \dots \end{bmatrix}$$

The matrices C_p , C_R , M_p , M_R have the following structure;

$$(4.10) \quad C_P = \begin{bmatrix} c_{p_1} \phi_1(s_1^y) & c_{p_1} \phi_2(s_1^y) & \dots & c_{p_1} \phi_N(s_1^y) \\ \vdots & & & \\ c_{p_k} \phi_1(s_k^y) & \dots & \dots & c_{p_k} \phi_N(s_k^y) \end{bmatrix}$$

$$(4.11) \quad C_R = \begin{bmatrix} c_{r_1} \phi_1(s_1^y) & c_{r_1} \phi_2(s_1^y) & \dots & c_{r_1} \phi_N(s_1^y) \\ \vdots & & & \\ c_{r_k} \phi_1(s_k^y) & \dots & \dots & \end{bmatrix}$$

$$(4.12) \quad M_P = \begin{bmatrix} m_{p_1} \phi_1(s_1^s) & m_{p_1} \phi_2(s_1^s) & \dots & m_{p_1} \phi_N(s_1^s) \\ \vdots & & & \\ m_{p_\ell} \phi_1(s_\ell^s) & \dots & \dots & \end{bmatrix}$$

$$(4.13) \quad M_R = \begin{bmatrix} m_{r_1} \phi_1(s_1^s) & m_{r_1} \phi_2(s_1^s) & \dots & m_{r_1} \phi_N(s_1^s) \\ \vdots & & & \\ m_{r_\ell} \phi_1(s_\ell^s) & \dots & \dots & \end{bmatrix}$$

The final point of interest concerning (4.5) is that it requires the exact eigenfunctions for the LSS. Even when \mathcal{D}_S is assumed equivalent to (4.3) and the LSS eigenfunctions become just the eigenfunctions of A_S the exact determination of these eigenfunctions is essentially impossible due to the size and complexity of the LSS structure. However, a very useful technique known as finite element analysis has been developed to provide an *estimate* for a finite number of

eigenfunctions at discrete points in the spatial domain ($\hat{\phi}_i(s_j)$, $i=1, \dots, N$, $j=1, \dots, \alpha$) and corresponding eigenvalues ($\hat{\lambda}_i$, $i=1, \dots, N$). The technique basically involves approximating the spatial domain S of the LSS by a finite number of meshes of particular shapes and well defined structural properties. These meshes are then patched together by a matching of the individual mesh boundary conditions, and the eigenfunction estimates at each of the nodes (mesh corners) are then derived. The number of eigenfunctions that can be estimated and the accuracy of the estimation is a direct function of the number and/or complexity of the meshes; while, the number of discrete points (α) at which the eigenfunctions are estimated is equal to the number of nodes. The finite element method is essentially a practical extension of the Galerkin numerical technique for approximating the solution of a partial differential equation. The method is particularly suited for use with a high-speed digital computer and computer programs such as NASTRAN have been used to provide approximate modal information for complex structures that defy analytical solutions.

The N approximate eigenfunction $\hat{\phi}_i(s_j)$ produced from finite element analysis can be directly substituted for the N exact mode shapes $\phi_i(s)$ used to develop (4.7) and this model can, with some degree of confidence, be used to represent the LSS for control design purposes. [73]

One last modification to the model of (4.7) is usually desirable. Since M is known to be a positive definite matrix and K is at least positive semi-definite a unitary transformation (T) can be applied to M and K which produces the following result:[74]

$$(4.14) \quad T^T M T = I_N \quad (\text{i.e. } N \times N \text{ identity matrix})$$

and

$$T^T K T = \Lambda \quad \text{where } \Lambda = \text{diag} (\omega_1^2, \dots, \omega_N^2)$$

where $\omega_1^2, \dots, \omega_N^2$ are the frequencies of modal motion. Therefore, using the coordinate transformation $q = T\eta$ (4.7) becomes:

$$(4.15a) \quad \left\{ \begin{array}{l} \ddot{\eta} + T^T D T \dot{\eta} + \Lambda \eta = T^T F u(t) \quad \eta \in R^N, \quad u \in R^m \\ y(t) = [C_p^T \quad C_R^T] \begin{bmatrix} \eta \\ \dot{\eta} \end{bmatrix} \quad y \in R^k \\ z(t) = [M_p^T \quad M_R^T] \begin{bmatrix} \eta \\ \dot{\eta} \end{bmatrix} \quad z \in R^l \end{array} \right.$$

where,

$$T^T D T = \text{diag} [2\zeta\omega_1 \quad \dots \quad 2\zeta\omega_N]$$

The second order model of (4.15a) can be readily converted to a state-space model of type $S(2N, k, m, l)$ by defining $x^T = [\eta^T \quad \dot{\eta}^T]$. This conversion is shown in (4.15b).

$$(4.15b) \quad \left\{ \begin{array}{l} \dot{x}(t) = Ax(t) + Bu(t) ; \quad x \in R^{2N}, \quad u \in R^m \\ y(t) = [C_p \quad C_R]Tx(t) ; \quad y \in R^k \\ z(t) = [M_p \quad M_R]Tx(t) ; \quad z \in R^l \end{array} \right.$$

where

$$A \triangleq \begin{bmatrix} 0 & I \\ -\Lambda & -T^T D T \end{bmatrix} \quad \text{and} \quad B \triangleq \begin{bmatrix} 0 \\ T^T F \end{bmatrix}$$

4.3 The Hoop Column Antenna

Figure 4.1 is a schematic of NASA's proposed hoop column antenna communications satellite.

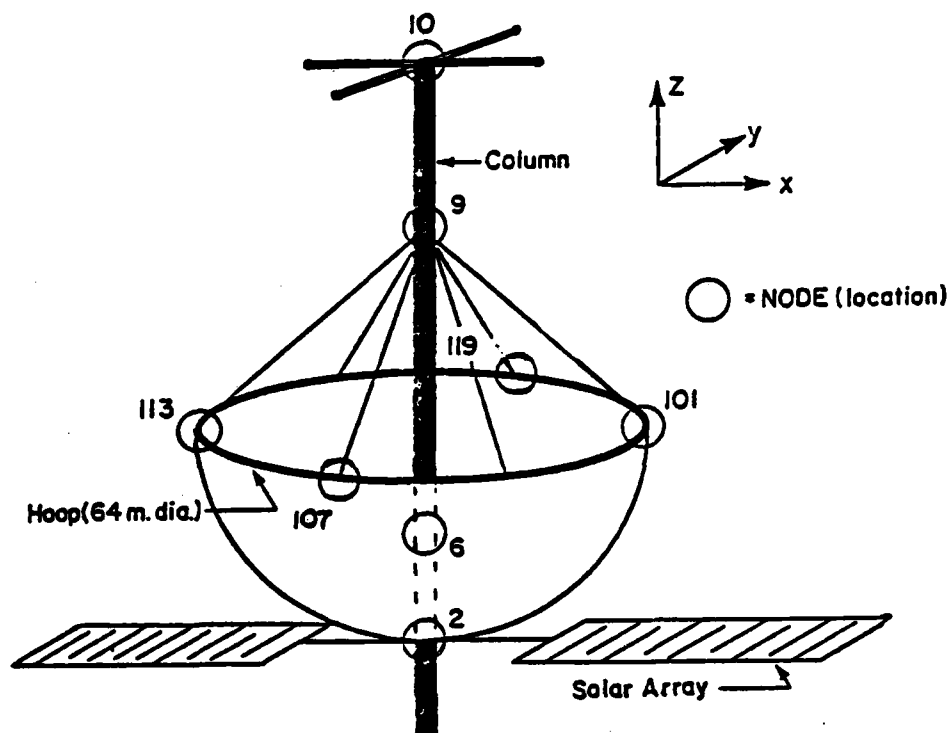


Figure 4.1: Hoop Column Antenna

The antenna will be positioned in a geosynchronous orbit with the mission of the control system being to maintain the focus and orientation of the satellite. The nodes shown on the schematic identify the spatial locations (s_i^y) which have been chosen to define 24 system outputs that are critical to the antenna focus and orientation. The nodes also identify the spatial locations (s_i^s) for 39 admissible sensors and the column nodes 2, 6, 9, 10 represent the spatial locations (s_i^a) for 12 admissible actuators. Tables 4.1-4.3 represent specific location, type and orientation information for the outputs and admissible sensors and actuators.

Table 4.1: Hoop Column Output Description

Output#	Type	Nodal Location	Direction
1	Inertial Angle	2	X
2	"	2	Y
3	"	2	Z
4	Relative Angle Between	10 and 2	X
5	"	"	Y
6	Inertial Angle	10	Z
7	Relative Linear Disp. Between	6 and 2	X
8	"	"	Y
9	"	9 and 2	X
10	"	"	Y
11	"	10 and 2	X
12	"	"	Y
13	"	101 and 10	X
14	"	"	Y
15	"	"	Z
16	"	107 and 10	X
17	"	"	Y
18	"	"	Z
19	"	113 and 10	X
20	"	"	Y
21	"	"	Z
22	"	119 and 10	X
23	"	"	Y
24	"	"	Z

Table 4.2: Hoop Column Sensor Description

Sensor#	Type	Nodal Location	Direction
1	Inertial Angle	2	X
2	"	"	Y
3	"	"	Z
4	Relative Linear Disp. Between	6 and 2	X
5	"	"	Y
6	"	"	Z
7	"	9 and 2	X
8	"	"	Y
9	"	"	Z
10	"	10 and 2	X
11	"	"	Y
12	"	"	Z
13	Inertial Angle	10	X
14	"	"	Y
15	"	"	Z
16	Relative Linear Disp. Between	101 and 10	X
17	"	"	Y
18	"	"	Z
19	"	107 and 10	X
20	"	"	Y
21	"	"	Z
22	"	113 and 10	X
23	"	"	Y
24	"	"	Z
25	"	119 and 10	X
26	"	"	Y
27	"	"	Z
28	Inertial Angular Rate	2	X
29	"	"	Y
30	"	"	Z
31	"	6	X
32	"	"	Y
33	"	"	Z
34	"	9	X
35	"	"	Y
36	"	"	Z
37	"	10	X
38	"	"	Y
39	"	"	Z

Table 4.3: Hoop Column Actuator Description

Actuator (Torquer)#	Nodal Location	Direction of Torque
1	2	X
2	2	Y
3	2	Z
4	6	X
5	6	Y
6	6	Z
7	9	X
8	9	Y
9	9	Z
10	10	X
11	10	Y
12	10	Z

The regulation specifications (σ^2 , μ^2) assumed for the hoop-column are defined in (4.16).

$$(4.16) \left\{ \begin{array}{ll} \sigma_i & \text{for linear displacements outputs} \quad .158 \text{ mm} \\ \sigma_i & \text{for angular outputs} \quad 22.8 \text{ }^\circ \\ \mu_i & \text{for all actuators} \quad 10 \text{ dn-cm} \end{array} \right.$$

4.3.1 Hoop Column NASTRAN Data and Model

The NASTRAN data used in this research to develop the hoop column model was generated by the Harris Corporation in early 1981. The data consisted of an estimate of the first 18 eigenfunctions for all 6 degrees of freedom in the hoop (i.e. translations and rotations in the coordinate directions x , y , z). The 6 degrees of freedom change the notation of section 4.2, and these changes are documented in Table 4.4.

Table 4.4: Multidimensional $\Omega(t,s)$ notation

$\Omega(t,s), \phi_i(s)$ become 6×1 vectors

m_{p_j}, m_{r_j} become $m_{p_j}^T, m_{r_j}^T$ (1×6 vectors)

c_p and c_r become $k \times 6$ matrices

c_{p_j}, c_{r_j} become $c_{p_j}^T, c_{r_j}^T$ (1×6 vectors)

f_i becomes a 6×1 vector

$\phi_j(s_i^a)$ becomes $\phi_j^T(s_i^a)$ a 1×6 vector

Also, the eigenfunction estimates were normalized by the mass density and this means that (4.7) automatically assumes the form of (4.15) without need of the transformation in (4.14). A sketch of the calculations necessary for selected elements of the matrices F, C_p, M_p , and M_R will be provided by first noting that C_p, M_p, M_R , and F can be written as follows:

$$C_p = \tilde{C}_p \phi(s^y) ;$$

$$(4.18a) \quad C_p = \begin{bmatrix} c_{p_1}^T & 0 & 0 & & \\ 0 & c_{p_2}^T & 0 & & \\ 0 & & & & \\ \vdots & & & & \\ & & & & c_{p_k}^T \end{bmatrix} \quad \phi(s^y) = \begin{bmatrix} \phi_1(s_1^y) & \dots & \phi_N(s_1^y) \\ \phi_1(s_k^y) & \dots & \dots \end{bmatrix}$$

and

$$(4.18b) \quad \phi_i(s_j^y) \triangleq [\phi_{i_x}(s_j^y), \phi_{i_y}(s_j^y), \phi_{i_z}(s_j^y), \phi_{i_{\theta_x}}(s_j^y), \phi_{i_{\theta_y}}(s_j^y), \phi_{i_{\theta}}(s_j^y)]^T,$$

where ϕ_i represents the linear shape function in the x direction, and $\phi_{i\theta_x}$ represents the angular shape function in the x direction.

$$(4.19) \quad M_p = \tilde{M}_p \phi(s^S) ; \quad M_R = \tilde{M}_R \phi(s^S)$$

where \tilde{M}_p and \tilde{M}_R have the same format as \tilde{C}_p with $c_{p_i}^T$ changed to $m_{p_i}^T$ or $m_{R_i}^T$ respectively and $\phi(s^S)$ is equivalent to $\phi(s^y)$ with output locations s^y changed to sensor locations s^S .

$$(4.20) \quad F = \phi^T(s^a) f ; \quad \text{where } f = \begin{bmatrix} f_1 \\ \vdots \\ f_m \end{bmatrix}$$

The matrix $\phi^T(s^a)$ is the transpose of $\phi(s^y)$ in (4.18) with s^y changed to s^a . It should be noted that no rate information (i.e. \dot{n}) is required in the hoop-column output vector and therefore $C_R = 0$. Table 4.5 displays the first four rows of C_p (i.e. $\tilde{c}_{p_i}^T$, $i = 1, 2, 3, 4$), the first four and last four rows of \tilde{M}_p and \tilde{M}_R (i.e. \tilde{m}_{p_i} and \tilde{m}_{R_i}), and the first 24 and last 6 entires of f . The first four rows of \tilde{C}_p correspond to the first four outputs defined in table (4.1). Similarly, the first four and last four rows of \tilde{M}_p and \tilde{M}_R correspond to the first four and last four measurements defined in table (4.2). The first 24 entries in f correspond to the first four actuators and the last 6 entires correspond to the last actuator as defined in table (4.3).

Combining the information provided in table 4.5 and (4.18)-(4.20) with the NASTRAN eigenfunction and eigenvalue data, a model of type

Table 4.5: Partial Representation of \tilde{C}_p , \tilde{M}_p , \tilde{M}_R , and f

$$\tilde{C}_{p1}^T = [000100 \quad 000000 \quad 000000 \quad 000000 \quad \dots \quad 000000]$$

$$\tilde{C}_{p2}^T = [000000 \quad 000010 \quad 000000 \quad 000000 \quad \dots \quad 000000]$$

$$\tilde{C}_{p3}^T = [000000 \quad 000000 \quad 000001 \quad 000000 \quad \dots \quad 000000]$$

$$\tilde{C}_{p4}^T = [000-100 \quad 000000 \quad 000000 \quad 000100 \quad \dots \quad 000000]$$

$$\tilde{m}_{p1}^T = [000100 \quad 000000 \quad 000000 \quad 000000 \quad \dots \quad 000000]$$

$$\tilde{m}_{p2}^T = [000000 \quad 000010 \quad 000000 \quad 000000 \quad \dots \quad 000000]$$

$$\tilde{m}_{p3}^T = [000000 \quad 000000 \quad 000001 \quad 000000 \quad \dots \quad 000000]$$

$$\tilde{m}_{p4}^T = [-100000 \quad 000000 \quad 000000 \quad 100000 \quad \dots \quad 000000]$$

$$\tilde{m}_{R1}^T - \tilde{m}_{R4}^T \quad \text{all zero rows}$$

$$\tilde{m}_{p36}^T - \tilde{m}_{p39}^T \quad \text{all zero rows}$$

$$\tilde{m}_{R36}^T = [000000 \quad \dots \quad 000001 \quad 000000 \quad 000000 \quad 000000]$$

$$\tilde{m}_{R37}^T = [000000 \quad \dots \quad 000000 \quad 000100 \quad 000000 \quad 000000]$$

$$\tilde{m}_{R38}^T = [000000 \quad \dots \quad 000000 \quad 000000 \quad 000010 \quad 000000]$$

$$\tilde{m}_{R39}^T = [000000 \quad \dots \quad 000000 \quad 000000 \quad 000000 \quad 000001]$$

$$f^T = [000100 \quad 000010 \quad 000001 \quad 0001000 \quad \dots \quad 000001]$$

(4.15b) was derived for the hoop-column where $N = 15$, $m = 12$, $k = 24$ and $\ell = 39$. (note: since the eigenfunctions were normalized, $T = 1$) Only 15 instead of the given 18 modes were used because the 3 translational rigid body modes were assumed to be non-excitable do to the satellites stable geosynchronous orbit. With this 15 mode model, it was discovered that two surface modes (labeled #14 and #15 by the Harris Corporation were unobservable and uncontrollable). Therefore, these modes were also deleted leaving a total of 13 modes (i.e. $N = 13$) in the hoop-column model.

For this 13 mode representation, a model of type $S(n,k,m,\ell)$ was generated by adding white noise processes to the model to account for sensor, actuator and model uncertainties. The form of the model is shown in (4.21):

$$(4.21) \left\{ \begin{array}{l} \dot{x}(t) = Ax(t) + Bu(t) + Dw(t) ; \quad x \in \mathbb{R}^{26} ; \quad u \in \mathbb{R}^{12} , \quad w \in \mathbb{R}^{24} \\ \\ D = [B \quad B] ; \quad (A,B) \text{ controllable} \\ \\ y(t) = Cx(t) ; \quad y \in \mathbb{R}^{24} ; \quad (A,C) \text{ observable} \\ \\ z(t) = Mx(t) + v(t) ; \quad z \in \mathbb{R}^{39} , \quad v \in \mathbb{R}^{39} ; \quad (A,M) \text{ measurable} \\ \\ E(w(t)) = 0 ; \quad E(v(t)) = 0 \\ \\ E \left(\begin{pmatrix} w(t) \\ v(t) \end{pmatrix} \begin{pmatrix} w^T(\tau) & v^T(\tau) \end{pmatrix} \right) = \begin{bmatrix} W\delta(t-\tau) & 0 \\ 0 & V\delta(t-\tau) \end{bmatrix} \end{array} \right.$$

It should be noted that although D is partitioned as $[B \quad B]$ the second partition is used to represent model error and will not change if the

number of actuators changes. The contents of the matrices A, B, C, M, W, and V are described below.

$$(4.22) \quad A = \left[\begin{array}{cccc} \overbrace{0 \quad I_{10} \quad 0 \quad 0}^{26} \\ \underbrace{-\omega^2}_{10 \times 10} \quad -2\xi\omega \quad 0 \quad 0 \\ 0 \quad 0 \quad 0 \quad I_3 \\ 0 \quad 0 \quad 0 \quad 0 \end{array} \right] \left. \vphantom{\begin{array}{c} 0 \\ -\omega^2 \\ 0 \\ 0 \end{array}} \right\} 26 ; \quad B = \left[\begin{array}{c} \overbrace{0}^{12} \\ BE \\ 0 \\ BR \end{array} \right] \left. \vphantom{\begin{array}{c} 0 \\ BE \\ 0 \\ BR \end{array}} \right\} 26$$

where

I_{10} , I_3 = 10x10 and 3x3 identity matrices

$-\omega^2$ = 10x10 diagonal matrix of the squared modal frequencies i.e.

$\omega^2 = \text{diag} [.40579, 7.2090, 7.2362, 13.277, 44.834, 132.14, 142.66, 445.01, 448.69, 775.86]$
(rad²/sec²)

$-2\xi\omega$ = 10x10 modal damping matrix, i.e.

$2\xi\omega = \text{diag} [.0127, .053699, .0538, .07286, .26283, .45981, .47777, .84381, .8473, 1.1142]$
(rads/sec)

$$(4.23) \quad C = [CE \quad 0 \quad CR \quad 0] \left. \vphantom{\begin{array}{c} CE \\ 0 \\ CR \\ 0 \end{array}} \right\} 24$$

$$(4.24) \quad M = \left[\begin{array}{cccc} \overbrace{ME \quad 0 \quad MR \quad 0}^{26} \\ 0 \quad MER \quad 0 \quad MRR \end{array} \right] \quad 39$$

The contents of the BE, BR, CE, CR, ME, MR, MER and MRR matrices are shown in tables 4.6-4.11. The noise intensity matrices are defined as follows:

$$(4.25) \quad W = \begin{bmatrix} W_1 & 0 \\ 0 & W_2 \end{bmatrix} ; \quad \begin{aligned} W_1 &= .1 I_{12} \text{ (dn-cm)}^2 \\ W_2 &= .1 \times 10^{-5} \text{ (dn-cm)}^2 \end{aligned}$$

$$(4.26) \quad V = \begin{bmatrix} V_1 & 0 & 0 & 0 & 0 \\ 0 & V_2 & 0 & 0 & 0 \\ 0 & 0 & V_1 & 0 & 0 \\ 0 & 0 & 0 & V_3 & 0 \\ 0 & 0 & 0 & 0 & V_4 \end{bmatrix} \left. \vphantom{\begin{bmatrix} V_1 & 0 & 0 & 0 & 0 \\ 0 & V_2 & 0 & 0 & 0 \\ 0 & 0 & V_1 & 0 & 0 \\ 0 & 0 & 0 & V_3 & 0 \\ 0 & 0 & 0 & 0 & V_4 \end{bmatrix}} \right\} 39 ; \quad \begin{aligned} V_1 &= 7.6154 \times 10^{-7} \text{ (rad)}^2 \\ V_2 &= 2.5 \times 10^{-7} I_9 \text{ (m}^2\text{)} \\ V_3 &= 2.5 \times 10^{-7} I_{12} \text{ (m}^2\text{)} \\ V_4 &= 4.7597 \times 10^{-5} I_{12} \text{ (rad/sec)}^2 \end{aligned}$$

4.3.2 The Hoop Column SASLQG Problem

The hoop column model defined by (4.21)-(4.26) and tables 4.6-4.11 will be labeled $S_{\text{Hoop}}(26,24,12,39)$. The following SASLQG problem is posed for hoop column.

Hoop Column SASLQG Problem (H.SAS Problem)

Given: $S_{\text{Hoop}}(26,24,12,39)$ with only 6 actuators and 12 sensors available for designing an LQG regulator to achieve the (σ^2, u^2) specifications of (4.16).

Required: Specify the closed-loop system which satisfies either the input constrained specifications of (3.20) or the output constrained specifications of (3.21).

The H.SAS problem will be one of the examples used to test the design algorithm of Chapter 7.

Table 4.6: The BE and BR Matrices

THE BE		MATRIX (10 BY 12)					
		1	2	3	4	5	6
1	-1.5580E-12	2.0391E-11	-1.1045E-02	-1.5494E-12	2.4122E-12	-1.0918E-02	
2	9.8045E-10	1.6688E-03	-4.9651E-12	9.8565E-10	1.6354E-03	-1.7175E-12	
3	-1.6599E-03	7.6073E-10	-1.0812E-12	-1.6433E-03	1.1361E-09	1.4252E-12	
4	-4.6650E-14	1.2781E-11	-1.3452E-02	-1.1506E-14	-8.4481E-12	4.0239E-03	
5	3.9018E-14	6.8800E-12	-7.2776E-03	-1.0665E-14	-4.5237E-12	2.4678E-02	
6	-1.9443E-10	7.6551E-03	9.1886E-13	9.8361E-12	-3.2844E-05	2.6411E-13	
7	-7.3458E-03	-4.9316E-10	1.6507E-13	-3.5997E-04	2.0989E-10	-1.1448E-13	
8	-2.0518E-11	2.8388E-03	3.1986E-12	2.2558E-12	-1.4116E-03	6.5322E-15	
9	-2.1316E-03	5.1427E-11	8.4627E-15	8.0639E-04	-3.4740E-11	5.2626E-15	
10	1.1447E-10	-1.4643E-02	-2.2370E-11	-8.9715E-14	1.0167E-02	2.5578E-15	
		7	8	9	10	11	12
1	-2.4578E-12	1.7996E-11	-1.0738E-02	-3.6592E-12	2.3622E-11	-1.0775E-02	
2	-7.7906E-10	-1.3052E-03	-4.6410E-13	-2.8930E-09	-4.8232E-03	-4.9025E-13	
3	1.3046E-03	-8.1944E-10	-2.8724E-13	4.8327E-03	-2.8561E-09	-6.8225E-13	
4	-2.9036E-14	2.2303E-12	2.1881E-02	-1.6287E-13	-9.8949E-13	2.4690E-02	
5	-9.2454E-14	1.2231E-12	-8.3798E-03	-4.6523E-13	-6.5593E-13	-1.3609E-02	
6	7.6056E-11	-3.3330E-03	5.4994E-15	-7.2756E-11	3.1574E-03	-3.9383E-14	
7	3.6726E-03	1.9284E-11	-1.0802E-13	-3.2853E-03	-3.6427E-11	3.7079E-14	
8	-3.9170E-13	5.5245E-04	-3.3571E-15	-1.8075E-13	9.1035E-04	9.1567E-16	
9	-2.3104E-05	9.4759E-12	3.6610E-14	-1.2039E-03	-4.8622E-12	1.2247E-15	
10	2.2502E-14	-9.3469E-03	7.4532E-15	1.3914E-14	4.5114E-03	-1.4097E-15	
THE BR		MATRIX (3 BY 12)					
		1	2	3	4	5	6
1	-6.5117E-04	8.8738E-08	3.3814E-11	-6.5118E-04	8.9906E-08	3.0492E-11	
2	8.9529E-08	6.5131E-04	2.3734E-10	8.9528E-08	6.5122E-04	2.3745E-10	
3	7.7548E-13	-1.2353E-11	6.0894E-03	7.6575E-13	-3.0140E-12	6.1338E-03	
		7	8	9	10	11	12
1	-7.1776E-04	9.8472E-08	3.1958E-11	-7.9365E-04	1.0907E-07	3.2475E-11	
2	9.8675E-08	7.1774E-04	2.3657E-10	1.0910E-07	7.9356E-04	2.3681E-10	
3	4.3753E-13	-5.2949E-12	6.2153E-03	1.6303E-14	-6.8254E-13	6.1993E-03	

Table 4.7: The CE Matrix

THE CE		MATRIX (24 BY 10)				
		1	2	3	4	5
1	-1.5980E-12	9.8045E-10	-1.6599E-03	-4.6650E-14	3.9018E-14	
2	2.0391E-11	1.6688E-03	7.6073E-10	1.2781E-11	6.8800E-12	
3	-1.1045E-02	-4.9651E-12	-1.0812E-12	-1.3452E-02	-7.2776E-03	
4	-2.0612E-12	-3.8735E-09	6.4926E-03	-1.1622E-13	-5.0425E-13	
5	3.2310E-12	-6.4920E-03	-3.6168E-09	-1.3770E-11	-7.5359E-12	
6	-1.0775E-02	-4.9025E-13	-6.8225E-13	2.4690E-02	-1.3609E-02	
7	1.1011E-10	2.3764E-02	1.4877E-08	-3.4696E-11	-1.8507E-11	
8	2.2688E-11	-1.4311E-08	2.3764E-02	3.7395E-13	-1.3531E-13	
9	3.7640E-10	4.3917E-02	2.6324E-08	-2.5766E-12	-2.9303E-13	
10	6.1152E-11	-2.6451E-08	4.4063E-02	5.0362E-13	-2.6354E-13	
11	7.3325E-10	-1.5985E-02	-9.4022E-09	-1.2017E-12	-7.7871E-13	
12	1.1462E-10	9.4569E-09	-1.5939E-02	2.4442E-12	5.3315E-12	
13	6.2644E-03	3.7333E-02	4.4234E-06	-2.6232E-04	2.7183E-03	
14	-1.0850E-02	4.8142E-06	3.7289E-02	4.5439E-04	-4.7082E-03	
15	4.1555E-08	1.7946E-02	1.0356E-02	-1.4920E-08	-1.8021E-08	
16	1.0850E-02	3.7330E-02	-4.8139E-06	-4.5439E-04	4.7082E-03	
17	6.2644E-03	-4.4221E-06	3.7293E-02	-2.6232E-04	2.7183E-03	
18	4.1399E-08	-1.0361E-02	1.7938E-02	-1.4919E-08	-1.8021E-08	
19	-4.1128E-04	3.7247E-02	-6.0024E-06	-3.4072E-04	-5.1321E-03	
20	7.1243E-04	-6.1945E-06	3.7214E-02	5.9011E-04	8.8890E-03	
21	5.7982E-08	-7.7130E-03	-4.4493E-03	-2.0848E-08	-2.5147E-08	
22	-7.1243E-04	3.7256E-02	5.7700E-06	-5.9011E-04	-8.8890E-03	
23	-4.1128E-04	6.4295E-06	3.7206E-02	-3.4072E-04	-5.1321E-03	
24	5.8020E-08	4.4525E-03	-7.7074E-03	-2.0881E-08	-2.5165E-08	
		6	7	8	9	10
1	-1.9443E-10	-7.3458E-03	-2.0518E-11	-2.1316E-03	1.1447E-10	
2	7.6551E-03	-4.9316E-10	2.8388E-03	5.1427E-11	-1.4643E-02	
3	9.1886E-13	1.6507E-13	3.1986E-12	8.4627E-15	-2.2370E-11	
4	1.2167E-10	4.0605E-03	2.0337E-11	9.2770E-04	-1.1446E-10	
5	-4.4977E-03	4.5673E-10	-1.9285E-03	-5.6289E-11	1.9154E-02	
6	-3.9383E-14	3.7079E-14	9.1567E-16	1.2247E-15	-1.4097E-15	
7	6.6063E-02	-6.3608E-10	7.4294E-03	-1.5739E-10	1.6830E-04	
8	1.5715E-09	6.8980E-02	2.9092E-10	9.8576E-03	-2.0204E-09	
9	-9.7582E-03	1.9237E-10	-1.4196E-02	-2.4179E-12	4.4242E-03	
10	-1.0048E-10	-8.8818E-03	2.8761E-10	-1.4117E-02	-2.0119E-09	
11	6.5518E-03	-1.0065E-10	-1.2157E-03	-3.7776E-12	2.6514E-03	
12	2.7943E-10	6.9694E-03	2.9173E-10	-8.2168E-04	-2.0122E-09	
13	5.1493E-02	-5.1044E-05	7.3548E-03	6.4842E-04	4.4588E-03	
14	-7.5693E-05	5.4072E-02	1.0799E-03	9.2917E-03	-1.0650E-04	
15	5.1511E-02	3.0689E-02	6.5278E-03	4.2838E-03	-1.1151E-02	
16	5.1524E-02	9.0562E-05	6.9553E-03	-1.0816E-03	4.6821E-03	
17	4.1696E-05	5.4033E-02	-6.4726E-04	9.6874E-03	-9.3106E-06	
18	-2.9739E-02	5.3157E-02	-3.7301E-03	7.4901E-03	6.5604E-03	
19	5.1311E-02	-1.7229E-04	7.3086E-03	3.3921E-04	3.2006E-03	
20	-2.0998E-04	5.3937E-02	5.7177E-04	9.3266E-03	5.0883E-04	
21	-2.2569E-02	-1.3332E-02	-2.6076E-03	-1.5122E-03	8.5958E-03	
22	5.1518E-02	1.9710E-04	7.1797E-03	-6.1042E-04	2.6757E-03	
23	1.8871E-04	5.3750E-02	-3.0086E-04	9.4965E-03	-5.8280E-04	
24	1.3029E-02	-2.3094E-02	1.4526E-03	-2.7153E-03	-5.1253E-03	

Table 4.8: The CR Matrix

THE CR		MATRIX (24 BY 3)		
		1	2	3
1	-6.5117E-04	8.9529E-08	7.7548E-13	
2	8.8738E-08	6.5131E-04	-1.2353E-11	
3	3.3814E-11	2.3734E-10	6.0894E-03	
4	-1.4248E-04	1.9571E-08	-7.5918E-13	
5	2.0332E-08	1.4225E-04	1.1670E-11	
6	3.2475E-11	2.3681E-10	6.1993E-03	
7	1.2908E-06	9.3852E-03	-8.0770E-11	
8	9.3833E-03	-1.2901E-06	-1.1069E-11	
9	3.2972E-06	2.4006E-02	-2.3233E-10	
10	2.4004E-02	-3.3002E-06	-2.6191E-11	
11	5.0290E-06	3.6613E-02	-2.6876E-10	
12	3.6612E-02	-5.0334E-06	-2.8763E-11	
13	-3.9225E-06	-2.8101E-02	5.8961E-03	
14	-2.8101E-02	3.8125E-06	-1.0212E-02	
15	4.0575E-03	7.0266E-03	-1.8068E-08	
16	-3.8071E-06	-2.8101E-02	1.0212E-02	
17	-2.8101E-02	3.9265E-06	5.8961E-03	
18	7.0257E-03	-4.0581E-03	-1.7958E-08	
19	-3.6891E-06	-2.8166E-02	-3.4692E-03	
20	-2.8167E-02	4.0548E-06	6.0088E-03	
21	-1.7160E-03	-2.9717E-03	-2.5163E-08	
22	-4.0539E-06	-2.8166E-02	-6.0088E-03	
23	-2.8167E-02	3.6969E-06	-3.4692E-03	
24	-2.9713E-03	1.7162E-03	-2.5195E-08	

Table 4.9: The ME Matrix

THE ME MATRIX (27 BY 10)

	1	2	3	4	5
1	-1.5980E-12	9.8045E-10	-1.6599E-03	-4.6650E-14	3.9018E-14
2	2.0391E-11	1.6688E-03	7.6073E-10	1.2781E-11	6.8800E-12
3	-1.1045E-02	-4.9651E-12	-1.0812E-12	-1.3452E-02	-7.2776E-03
4	1.1011E-10	2.3764E-02	1.4877E-08	-3.4696E-11	-1.8507E-11
5	2.2688E-11	-1.4311E-08	2.3764E-02	3.7395E-13	-1.3531E-13
6	-1.8000E-14	6.8493E-10	-7.8900E-15	3.4700E-15	7.2520E-15
7	3.7640E-10	4.3917E-02	2.6324E-08	-2.5766E-12	-2.9303E-13
8	6.1152E-11	-2.6451E-08	4.4063E-02	5.0362E-13	-2.6354E-13
9	-3.7000E-14	-3.3290E-12	6.4320E-14	9.5700E-15	1.6079E-14
10	7.3325E-10	-1.5985E-02	-9.4022E-09	-1.2017E-12	-7.7871E-13
11	1.1462E-10	9.4569E-09	-1.5939E-02	2.4442E-12	5.3315E-12
12	-5.1000E-14	-3.3120E-12	5.5790E-14	1.6350E-14	2.4573E-14
13	-3.6592E-12	-2.8930E-09	4.8327E-03	-1.6287E-13	-4.6523E-13
14	2.3622E-11	-4.8232E-03	-2.8561E-09	-9.8949E-13	-6.5593E-13
15	-1.0775E-02	-4.9025E-13	-6.8225E-13	2.4690E-02	-1.3609E-02
16	6.2644E-03	3.7333E-02	4.4234E-06	-2.6232E-04	2.7183E-03
17	-1.0850E-02	4.8142E-06	3.7289E-02	4.5439E-04	-4.7082E-03
18	4.1555E-08	1.7946E-02	1.0356E-02	-1.4920E-08	-1.8021E-08
19	1.0850E-02	3.7330E-02	-4.8139E-06	-4.5439E-04	4.7082E-03
20	6.2644E-03	-4.4221E-06	3.7293E-02	-2.6232E-04	2.7183E-03
21	4.1399E-08	-1.0361E-02	1.7938E-02	-1.4919E-08	-1.8021E-08
22	-4.1128E-04	3.7247E-02	-6.0024E-06	-3.4072E-04	-5.1321E-03
23	7.1243E-04	-6.1945E-06	3.7214E-02	5.9011E-04	8.8890E-03
24	5.7982E-08	-7.7130E-03	-4.4493E-03	-2.0848E-08	-2.5147E-08
25	-7.1243E-04	3.7256E-02	5.7700E-06	-5.9011E-04	-8.8890E-03
26	-4.1128E-04	6.4295E-06	3.7206E-02	-3.4072E-04	-5.1321E-03
27	5.8020E-08	4.4525E-03	-7.7074E-03	-2.0881E-08	-2.5165E-08
	6	7	8	9	10
1	-1.9443E-10	-7.3458E-03	-2.0518E-11	-2.1316E-03	1.1447E-10
2	7.6551E-03	-4.9316E-10	2.8388E-03	5.1427E-11	-1.4643E-02
3	9.1886E-13	1.6507E-13	3.1986E-12	8.4627E-15	-2.2370E-11
4	6.6063E-02	-6.3608E-10	7.4294E-03	-1.5739E-10	1.6830E-04
5	1.5715E-09	6.8980E-02	2.9092E-10	9.8576E-03	-2.0204E-09
6	3.1619E-09	-8.4540E-14	6.0273E-10	-8.2300E-14	-8.4809E-10
7	-9.7582E-03	1.9237E-10	-1.4196E-02	-2.4179E-12	4.4242E-03
8	-1.0048E-10	-8.8818E-03	2.8761E-10	-1.4117E-02	-2.0119E-09
9	-3.2403E-11	-2.0090E-14	-1.1865E-11	-3.6000E-15	5.2664E-11
10	6.5518E-03	-1.0065E-10	-1.2157E-03	-3.7776E-12	2.6514E-03
11	2.7943E-10	6.9694E-03	2.9173E-10	-8.2168E-04	-2.0122E-09
12	-3.2405E-11	-5.0430E-14	-1.1977E-11	-1.7610E-13	5.2646E-11
13	-7.2756E-11	-3.2853E-03	-1.8075E-13	-1.2039E-03	1.3914E-14
14	3.1574E-03	-3.6427E-11	9.1035E-04	-4.8622E-12	4.5114E-03
15	-3.9383E-14	3.7079E-14	9.1567E-16	1.2247E-15	-1.4097E-15
16	5.1493E-02	-5.1044E-05	7.3548E-03	6.4842E-04	4.4588E-03
17	-7.5693E-05	5.4072E-02	1.0799E-03	9.2917E-03	-1.0650E-04
18	5.1511E-02	3.0689E-02	6.5278E-03	4.2838E-03	-1.1151E-02
19	5.1524E-02	9.0562E-05	6.9553E-03	-1.0816E-03	4.6821E-03
20	4.1696E-05	5.4033E-02	-6.4726E-04	9.6874E-03	-9.3106E-06
21	-2.9739E-02	5.3157E-02	-3.7301E-03	7.4901E-03	6.5604E-03
22	5.1311E-02	-1.7229E-04	7.3086E-03	3.3921E-04	3.2006E-03
23	-2.0998E-04	5.3937E-02	5.7177E-04	9.3266E-03	5.0883E-04
24	-2.2569E-02	-1.3332E-02	-2.6076E-03	-1.5122E-03	8.5958E-03
25	5.1518E-02	1.9710E-04	7.1797E-03	-6.1042E-04	2.6757E-03
26	1.8871E-04	5.3750E-02	-3.0086E-04	9.4965E-03	-5.8280E-04
27	1.3029E-02	-2.3094E-02	1.4526E-03	-2.7153E-03	-5.1253E-03

Table 4.10: The MR Matrix

THE MR		MATRIX (27 BY 3)		
		1	2	3
1	-6.5117E-04	8.9529E-08	7.7548E-13	
2	8.8738E-08	6.5131E-04	-1.2353E-11	
3	3.3814E-11	2.3734E-10	6.0894E-03	
4	1.2908E-06	9.3852E-03	-8.0770E-11	
5	9.3833E-03	-1.2901E-06	-1.1069E-11	
6	-1.0000E-13	-2.0928E-10	7.0000E-15	
7	3.2972E-06	2.4006E-02	-2.3233E-10	
8	2.4004E-02	-3.3002E-06	-2.6191E-11	
9	5.0000E-14	2.7900E-12	2.1000E-14	
10	5.0290E-06	3.6613E-02	-2.6876E-10	
11	3.6612E-02	-5.0334E-06	-2.8763E-11	
12	3.0000E-14	2.7900E-12	2.9000E-14	
13	-7.9365E-04	1.0910E-07	1.6303E-14	
14	1.0907E-07	7.9356E-04	-6.8254E-13	
15	3.2475E-11	2.3681E-10	6.1993E-03	
16	-3.9225E-06	-2.8101E-02	5.8961E-03	
17	-2.8101E-02	3.8125E-06	-1.0212E-02	
18	4.0575E-03	7.0266E-03	-1.8068E-08	
19	-3.8071E-06	-2.8101E-02	1.0212E-02	
20	-2.8101E-02	3.9265E-06	5.8961E-03	
21	7.0257E-03	-4.0581E-03	-1.7958E-08	
22	-3.6891E-06	-2.8166E-02	-3.4692E-03	
23	-2.8167E-02	4.0548E-06	6.0088E-03	
24	-1.7160E-03	-2.9717E-03	-2.5163E-08	
25	-4.0539E-06	-2.8166E-02	-6.0088E-03	
26	-2.8167E-02	3.6969E-06	-3.4692E-03	
27	-2.9713E-03	1.7162E-03	-2.5195E-08	

Table 4.11: The MER and MRR Matrices

 THE MER MATRIX (12 BY 10)

	1	2	3	4	5
1	-1.5980E-12	9.8045E-10	-1.6599E-03	-4.6650E-14	3.9018E-14
2	2.0391E-11	1.6688E-03	7.6073E-10	1.2781E-11	6.8800E-12
3	-1.1045E-02	-4.9651E-12	-1.0812E-12	-1.3452E-02	-7.2776E-03
4	-1.5494E-12	9.8565E-10	-1.6433E-03	-1.1506E-14	-1.0665E-14
5	2.4122E-12	1.6354E-03	1.1361E-09	-8.4481E-12	-4.5237E-12
6	-1.0918E-02	-1.7175E-12	1.4252E-12	4.0239E-03	2.4678E-02
7	-2.4578E-12	-7.7906E-10	1.3046E-03	-2.9036E-14	-9.2454E-14
8	1.7996E-11	-1.3052E-03	-8.1944E-10	2.2303E-12	1.2231E-12
9	-1.0738E-02	-4.6410E-13	-2.8724E-13	2.1881E-02	-8.3798E-03
10	-3.6592E-12	-2.8930E-09	4.8327E-03	-1.6287E-13	-4.6523E-13
11	2.3622E-11	-4.8232E-03	-2.8561E-09	-9.8949E-13	-6.5593E-13
12	-1.0775E-02	-4.9025E-13	-6.8225E-13	2.4690E-02	-1.3609E-02

	6	7	8	9	10
1	-1.9443E-10	-7.3458E-03	-2.0518E-11	-2.1316E-03	1.1447E-10
2	7.6551E-03	-4.9316E-10	2.8388E-03	5.1427E-11	-1.4643E-02
3	9.1866E-13	1.6507E-13	3.1986E-12	8.4627E-15	-2.2370E-11
4	9.8361E-12	-3.5997E-04	2.2558E-12	8.0639E-04	-8.9715E-14
5	-3.2844E-05	2.0989E-10	-1.4116E-03	-3.4740E-11	1.0167E-02
6	2.6411E-13	-1.1448E-13	6.5322E-15	5.2626E-15	2.5578E-15
7	7.6056E-11	3.6726E-03	-3.9170E-13	-2.3104E-05	2.2502E-14
8	-3.3330E-03	1.9284E-11	5.5245E-04	9.4759E-12	-9.3469E-03
9	5.4994E-15	-1.0802E-13	-3.3571E-15	3.6610E-14	7.4532E-15
10	-7.2756E-11	-3.2853E-03	-1.8075E-13	-1.2039E-03	1.3914E-14
11	3.1574E-03	-3.6427E-11	9.1035E-04	-4.8622E-12	4.5114E-03
12	-3.9383E-14	3.7079E-14	9.1567E-16	1.2247E-15	-1.4097E-15

 THE MRR MATRIX (12 BY 3)

	1	2	3
1	-6.5117E-04	8.9529E-08	7.7548E-13
2	8.8738E-08	6.5131E-04	-1.2353E-11
3	3.3814E-11	2.3734E-10	6.0894E-03
4	-6.5118E-04	8.9528E-08	7.6575E-13
5	8.9906E-08	6.5122E-04	-3.0140E-12
6	3.0492E-11	2.3745E-10	6.1338E-03
7	-7.1776E-04	9.8675E-08	4.3753E-13
8	9.8472E-08	7.1774E-04	-5.2949E-12
9	3.1958E-11	2.3657E-10	6.2153E-03
10	-7.9365E-04	1.0910E-07	1.6303E-14
11	1.0907E-07	7.9356E-04	-6.8254E-13
12	3.2475E-11	2.3681E-10	6.1993E-03

4.4 The Solar Optical Telescope

Figure 4.2 is a schematic of the solar optical telescope.

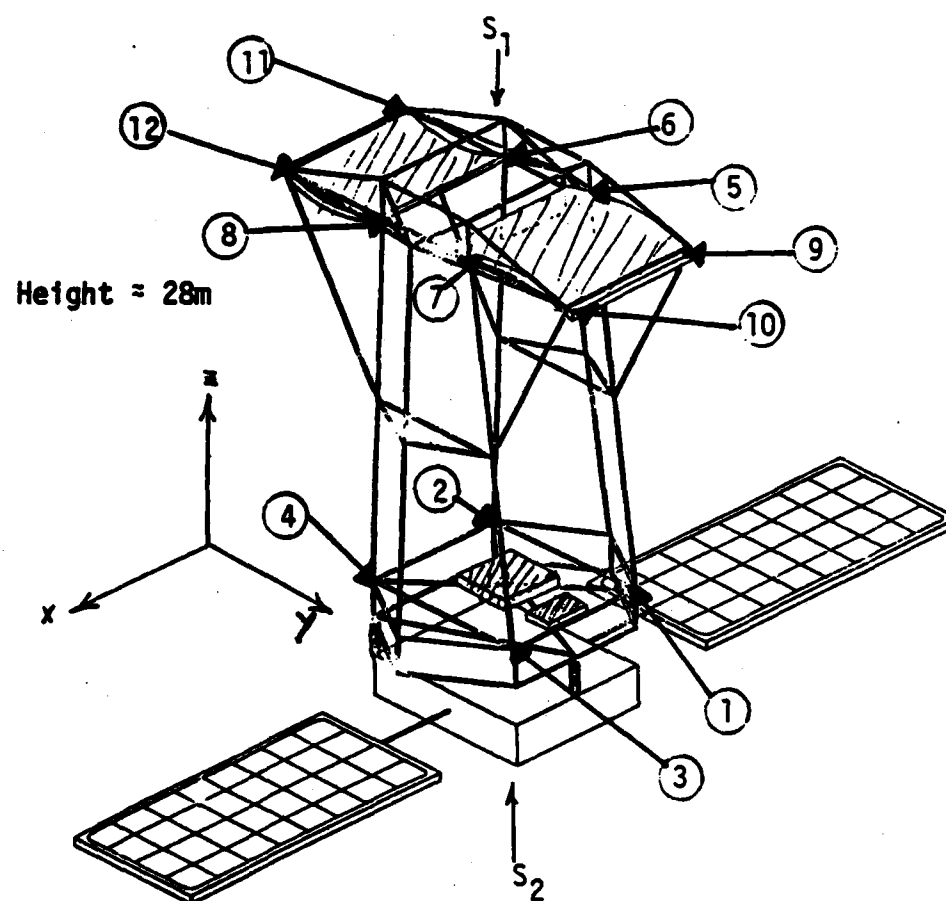


Figure 4.2: The Solar Optical Telescope

The telescope model was developed by the Charles Stark Draper Laboratory (CSDL) primarily for the purpose of providing a minimum complexity structure for the evaluation of LSS control design techniques. The outputs for the model (not shown in Fig. 4.2) are the telescope line of sight in the x and y directions and the focal length (defocus) of the lenses located at the top and bottom of the telescope. The nodes shown on the telescope represent the spatial locations for 45 admissible sensors and 21 admissible actuators. Tables 4.12-4.14 provide

Table 4.12: Telescope Sensor Description

Sensor#	Type	Nodal Location	Direction
1	Line of sight angle	---	X
2	"	---	Y
3	Defocus	---	---
4	Linear Displacement	1	Y
5	"	1	Z
6	"	2	Z
7	"	3	X
8	"	3	Y
9	"	3	Z
10	"	4	Z
11	"	5	X
12	"	5	Y
13	"	5	Z
14	"	6	Z
15	"	7	Y
16	"	7	Z
17	"	8	Z
18	"	9	Z
19	"	10	Z
20	"	11	X
21	"	11	Y
22	"	11	Z
23	"	12	Y
24	"	12	Z
25	Linear Rate	1	Y
26	"	1	Z
27	"	2	Z
28	"	3	X
29	"	3	Y
30	"	3	Z
31	"	4	Z
32	"	5	X
33	"	5	Y
34	"	5	Z
35	"	6	Z
36	"	7	Y
37	"	7	Z
38	"	8	Z
39	"	9	Z
40	"	10	Z
41	"	11	X
42	"	11	Y
43	"	11	Z
44	"	12	Y
45	"	12	Z

Table 4.13: Telescope Actuator Description

Actuator (Force)#	Nodal Location	Direction
1	1	Y
2	1	Z
3	2	Z
4	3	X
5	3	Y
6	3	Z
7	4	Z
8	5	X
9	5	Y
10	5	Z
11	6	Z
12	7	Y
13	7	Z
14	8	Z
15	9	Z
16	10	Z
17	11	X
18	11	Y
19	11	Z
20	12	Y
21	12	Z

Table 4.14: Telescope Specifications

		Specification
σ_1	Optical line of sight angle (LOS_x)	65.2 $\widehat{\text{sec}}$
σ_2	Optical line of sight angle (LOS_y)	65.2 $\widehat{\text{sec}}$
σ_3	defocus	.001 mm
μ_1	force actuator	.01 N

specific location, type and orientation information, for these admissible sensor and actuators and also a listing of the specifications (σ^2 , μ^2) chosen for the outputs and actuators.

4.4.1 Telescope Model Development

The telescope model used in this research was developed from NASTRAN data generated by CSDL in early 1980. The data contained estimates for the first 44 mode shapes of the structure and also provided location information for the two sinusoidal disturbances S_1 , S_2 shown in figure 4.2. Using the technique discussed in Section 4.2, a 10 mode, 20 state, linear stochastic model coupled with a 2 mode, 4 state, linear stochastic model of the disturbances were adopted. The criterion used for choosing 10 modes from the 44 modes was the component cost algorithm developed in [4]-[7]. The technique ranks the modes based upon their contribution to a quadratic functional of the system outputs. A discussion of this modal cost analysis selection algorithm as applied to the CSDL NASTRAN data is provided in [75]. The end result is a system of type S(24, 3, 21, 45) described below:

$$(4.27) \left\{ \begin{array}{l} \dot{x}(t) = Ax(t) + Bu(t) + Dw(t) ; \quad x \in R^{24}, \quad u \in R^{21}, \quad w \in R^{23} \\ \hspace{15em} (A,B) \text{ controllable} \\ y(t) = Cx(t) ; \quad y \in R^3 ; \quad (A,C) \text{ observable} \\ z(t) = Mx(t) + v(t) ; \quad z, v \in R^{45} ; \quad (A,M) \text{ measurable} \\ E\{w(t)\} = 0, \quad E\{v(t)\} = 0 \\ E \left\{ \begin{pmatrix} w(t) \\ v(t) \end{pmatrix} \begin{pmatrix} w^T(\tau) & v^T(\tau) \end{pmatrix} \right\} = \begin{bmatrix} \overline{W}\delta(t-\tau) & 0 \\ 0 & V\delta(t-\tau) \end{bmatrix} \end{array} \right.$$

The contents of the matrices A, B, C, D, M, W and V are described as follows:

$$(4.28) \quad A = \begin{bmatrix} 0 & I_8 & 0 & 0 & 0 & 0 \\ -\omega^2 & -2\xi\omega & S & 0 & 0 & 0 \\ 0 & 0 & 0 & I_2 & 0 & 0 \\ 0 & 0 & -\omega_s^2 & -2\xi\omega_s & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I_2 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \left. \vphantom{\begin{bmatrix} 0 & I_8 & 0 & 0 & 0 & 0 \\ -\omega^2 & -2\xi\omega & S & 0 & 0 & 0 \\ 0 & 0 & 0 & I_2 & 0 & 0 \\ 0 & 0 & -\omega_s^2 & -2\xi\omega_s & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I_2 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}} \right\} 24 ; \quad B = \begin{bmatrix} 0 \\ BE \\ 0 \\ BR \end{bmatrix} \quad \left. \vphantom{\begin{bmatrix} 0 \\ BE \\ 0 \\ BR \end{bmatrix}} \right\} 24$$

where S is the coupling matrix between the telescope and the disturbances S_1 and S_2 and,

$I_8, I_2 = 8 \times 8$ and 2×2 identity matrices

$-\omega^2 = 8 \times 8$ matrix of the squared modal frequencies, i.e.

$\omega^2 = \text{diag. } [.8347, 2.7356, 3.9706, 4.3776, 7.7455, 13.175, 13.339, 59.112] \text{ (rad}^2/\text{sec}^2)$

$-2\xi\omega = 8 \times 8$ modal damping matrix, i.e.

$2\xi\omega = \text{diag } [.001827, .0033079, .003985, .004184, .005566, .007259, .0073046, .015378] \text{ (rad/sec)}$

$-\omega_s^2 = 2 \times 2$ matrix of the squared disturbance frequencies, i.e.

$\omega_s^2 = \text{diag } 1 [3947.8, 986.96] \text{ (rad}^2/\text{sec}^2)$

$-2\xi\omega_s = 2 \times 2$ disturbance damping matrix, i.e.

$2s\omega_s = \text{diag } [.1257, .0628] \text{ (rad/sec)}$

$$(4.29) \quad D = \left[\begin{array}{cc} \overbrace{\begin{matrix} 0 & 0 \\ BE & 0 \\ 0 & 0 \\ 0 & I_2 \\ 0 & 0 \\ BR & 0 \end{matrix}}^{23} \end{array} \right] \left. \vphantom{\begin{matrix} 0 \\ BE \\ 0 \\ 0 \\ 0 \\ BR \end{matrix}} \right\} 24$$

$$(4.30) \quad C = \left[\overbrace{\begin{matrix} CE & 0 & 0 & CR & 0 \end{matrix}}^{24} \right] \} 3$$

$$(4.31) \quad M = \left[\begin{array}{ccccc} \overbrace{\begin{matrix} CE & 0 & 0 & CR & 0 \\ ME & 0 & 0 & MR & 0 \\ 0 & ME & 0 & 0 & MRR \end{matrix}}^{24} \end{array} \right] \} 45$$

The contents of the matrices S, BE, BR, CE, CR, ME, and MR are shown in tables 4.15-4.18.

The noise intensity matrices are defined as follows:

$$(4.32) \quad W = \left[\begin{array}{cc} \overbrace{\begin{matrix} W_1 & 0 \\ 0 & W_2 \end{matrix}}^{23} \end{array} \right] \} 23 ; \quad \begin{aligned} W_1 &= (0.1) I_{21} (N^2) \\ W_2 &= (3.95) I_2 (N^2) \end{aligned}$$

$$(4.33) \quad V = \left[\begin{array}{ccc} \overbrace{\begin{matrix} V_1 & 0 & 0 \\ 0 & V_2 & 0 \\ 0 & 0 & V_3 \end{matrix}}^{45} \end{array} \right] \} 45 ; \quad \begin{aligned} V_1 &= (1.0 \times 10^{-4}) I_2 \text{ (rad)}^2 \\ V_2 &= (1.0 \times 10^{-6}) I_{22} \text{ (m)}^2 \\ V_3 &= (1.0 \times 10^{-7}) I_{21} \text{ (m/s)}^2 \end{aligned}$$

Table 4.15: The S and BE Matrices

THE S		MATRIX (8 BY 2)	
1	-1.5105E-03	6.1789E-03	
2	8.0569E-04	-9.4151E-05	
3	-5.1719E-03	-7.4586E-04	
4	1.0263E-03	1.4033E-03	
5	7.8788E-03	-5.0280E-03	
6	-3.0956E-03	-4.2834E-03	
7	-3.5394E-03	-5.3071E-04	
8	1.3823E-04	2.2141E-02	

THE BE		MATRIX (8 BY 21)					
1	6.7398E-05	-1.5203E-03	-1.5071E-03	1.9461E-03	1.5206E-03	1.5068E-03	7
2	-4.6154E-03	9.3534E-04	7.9167E-04	-7.8443E-03	4.6117E-03	-9.3542E-04	-7.9106E-04
3	-2.5821E-04	-5.1396E-03	-5.1753E-03	2.3592E-06	-2.5814E-04	-5.1395E-03	-5.1753E-03
4	-4.9274E-03	-7.8871E-04	1.4790E-03	1.0836E-06	-4.9299E-03	-7.8891E-04	1.4799E-03
5	2.7860E-04	7.7980E-03	7.8853E-03	-7.1741E-06	2.7837E-04	7.7978E-03	7.8853E-03
6	1.5566E-02	2.6261E-03	-4.5157E-03	-6.1490E-05	1.5583E-02	2.6072E-03	-4.5461E-03
7	-3.7921E-04	-3.4506E-03	-3.5444E-03	1.7048E-02	4.8135E-04	3.4683E-03	3.5145E-03
8	-6.0651E-05	8.1133E-05	1.4746E-04	4.8504E-04	6.0648E-05	-8.1247E-05	-1.4714E-04

1	-6.0352E-03	8.6611E-05	-1.5172E-03	-1.5096E-03	1.5172E-03	1.5096E-03	14
2	-1.5745E-03	-4.9430E-03	9.1748E-04	8.0491E-04	4.9415E-03	-9.1803E-04	-8.0496E-04
3	-3.4857E-08	-3.2804E-04	-5.1520E-03	-5.1720E-03	-3.2895E-04	-5.1520E-03	-5.1720E-03
4	-4.4647E-09	-3.8926E-04	-3.3665E-04	1.0263E-03	-3.8960E-04	-3.3656E-04	1.0270E-03
5	1.0398E-07	4.4976E-04	7.8314E-03	7.8791E-03	4.5023E-04	7.8314E-03	7.8791E-03
6	2.3112E-06	1.2568E-03	1.2122E-03	-3.0951E-03	1.2500E-03	1.1903E-03	-3.1239E-03
7	-4.7891E-04	-5.9153E-04	-3.4728E-03	-3.5398E-03	5.9857E-04	3.4807E-03	3.5190E-03
8	-3.0856E-04	6.4021E-05	9.5121E-05	1.4169E-04	-6.4624E-05	-9.5518E-05	-1.4084E-04

1	-1.5215E-03	1.5219E-03	-6.3131E-03	8.5420E-05	-1.5036E-03	-8.5364E-05	1.5030E-03	21
2	1.0421E-03	-1.0431E-03	1.4522E-02	-4.9453E-04	6.9308E-04	4.9448E-03	-6.9042E-04	
3	-5.1323E-03	-5.1322E-03	2.5935E-07	-3.2895E-04	5.1946E-03	-3.2902E-04	-5.1947E-03	
4	-1.9280E-03	-1.9285E-03	-3.9064E-07	-3.8315E-04	2.6169E-03	-3.8942E-04	2.6184E-03	
5	7.7864E-03	7.7858E-03	-5.7021E-07	4.5009E-04	7.9327E-03	4.5043E-04	7.9330E-03	
6	6.2495E-03	6.2355E-03	-1.0374E-05	1.2550E-03	-8.1253E-03	1.2482E-03	-8.1624E-03	
7	-3.4045E-03	3.4456E-03	1.4729E-03	-5.9276E-04	-3.6105E-03	6.0083E-04	3.5576E-03	
8	6.0750E-05	-6.1233E-05	-5.0078E-04	5.9931E-05	1.8241E-04	-5.9368E-05	-1.8238E-04	

Table 4.16: The BR, CE and CR Matrices

THE BR																					MATRIX (2 BY 21)																				
1	4.9314E-06	3.9075E-06	-4.8486E-06	0.	4.9314E-06	3.9075E-06	-4.8486E-06	0.	4.9314E-06	3.9075E-06	-4.8486E-06	0.	4.9314E-06	3.9075E-06	-4.8486E-06	0.	4.9314E-06	3.9075E-06	-4.8486E-06	0.	4.9314E-06																				
2	-4.5756E-07	3.2730E-06	3.2730E-06	-5.1189E-06	-4.5756E-07	3.2730E-06	3.2730E-06	-5.1189E-06	-4.5756E-07	3.2730E-06	3.2730E-06	-5.1189E-06	-4.5756E-07	3.2730E-06	3.2730E-06	-5.1189E-06	-4.5756E-07	3.2730E-06	3.2730E-06	-5.1189E-06	-4.5756E-07																				
1	0.	-1.2581E-05	2.1563E-06	-3.0974E-06	-1.2581E-05	2.1563E-06	-3.0974E-06	-1.2581E-05	2.1563E-06	-3.0974E-06	-1.2581E-05	2.1563E-06	-3.0974E-06	-1.2581E-05	2.1563E-06	-3.0974E-06	-1.2581E-05	2.1563E-06	-3.0974E-06	-1.2581E-05	2.1563E-06																				
2	1.1475E-05	-4.5756E-07	3.2730E-06	3.2730E-06	1.1475E-05	-4.5756E-07	3.2730E-06	3.2730E-06	1.1475E-05	-4.5756E-07	3.2730E-06	3.2730E-06	1.1475E-05	-4.5756E-07	3.2730E-06	3.2730E-06	1.1475E-05	-4.5756E-07	3.2730E-06	3.2730E-06	1.1475E-05																				
1	8.2855E-06	8.2855E-06	0.	-1.2581E-05	8.2855E-06	8.2855E-06	0.	-1.2581E-05	8.2855E-06	8.2855E-06	0.	-1.2581E-05	8.2855E-06	8.2855E-06	0.	-1.2581E-05	8.2855E-06	8.2855E-06	0.	-1.2581E-05	8.2855E-06																				
2	3.2730E-06	-3.2730E-06	1.2962E-05	-4.5756E-07	3.2730E-06	-3.2730E-06	1.2962E-05	-4.5756E-07	3.2730E-06	-3.2730E-06	1.2962E-05	-4.5756E-07	3.2730E-06	-3.2730E-06	1.2962E-05	-4.5756E-07	3.2730E-06	-3.2730E-06	1.2962E-05	-4.5756E-07	3.2730E-06																				
THE CE																					MATRIX (3 BY 8)																				
1	-2.5320E-07	-3.4775E-07	1.2949E-06	2.2830E-04	-2.5320E-07	-3.4775E-07	1.2949E-06	2.2830E-04	-2.5320E-07	-3.4775E-07	1.2949E-06	2.2830E-04	-2.5320E-07	-3.4775E-07	1.2949E-06	2.2830E-04	-2.5320E-07	-3.4775E-07	1.2949E-06	2.2830E-04	-2.5320E-07																				
2	3.2526E-04	2.1451E-04	3.4710E-07	-4.4224E-07	3.2526E-04	2.1451E-04	3.4710E-07	-4.4224E-07	3.2526E-04	2.1451E-04	3.4710E-07	-4.4224E-07	3.2526E-04	2.1451E-04	3.4710E-07	-4.4224E-07	3.2526E-04	2.1451E-04	3.4710E-07	-4.4224E-07	3.2526E-04																				
3	-5.0254E-08	1.0999E-06	5.2149E-06	2.7611E-06	-5.0254E-08	1.0999E-06	5.2149E-06	2.7611E-06	-5.0254E-08	1.0999E-06	5.2149E-06	2.7611E-06	-5.0254E-08	1.0999E-06	5.2149E-06	2.7611E-06	-5.0254E-08	1.0999E-06	5.2149E-06	2.7611E-06	-5.0254E-08																				
1	6.0324E-06	-7.3379E-04	-2.2959E-06	1.0205E-06	6.0324E-06	-7.3379E-04	-2.2959E-06	1.0205E-06	6.0324E-06	-7.3379E-04	-2.2959E-06	1.0205E-06	6.0324E-06	-7.3379E-04	-2.2959E-06	1.0205E-06	6.0324E-06	-7.3379E-04	-2.2959E-06	1.0205E-06	6.0324E-06																				
2	1.0661E-06	6.7350E-06	-8.7265E-04	2.3598E-04	1.0661E-06	6.7350E-06	-8.7265E-04	2.3598E-04	1.0661E-06	6.7350E-06	-8.7265E-04	2.3598E-04	1.0661E-06	6.7350E-06	-8.7265E-04	2.3598E-04	1.0661E-06	6.7350E-06	-8.7265E-04	2.3598E-04	1.0661E-06																				
3	1.5515E-05	-1.0160E-05	4.0611E-07	4.9916E-08	1.5515E-05	-1.0160E-05	4.0611E-07	4.9916E-08	1.5515E-05	-1.0160E-05	4.0611E-07	4.9916E-08	1.5515E-05	-1.0160E-05	4.0611E-07	4.9916E-08	1.5515E-05	-1.0160E-05	4.0611E-07	4.9916E-08	1.5515E-05																				
THE CR																					MATRIX (3 BY 2)																				
1	1.0000E+00	0.	2	1.0000E+00	1.0000E+00	0.	2	1.0000E+00	1.0000E+00	0.	2	1.0000E+00	1.0000E+00	0.	2	1.0000E+00	1.0000E+00	0.	2	1.0000E+00	1.0000E+00																				
2	0.	1.0000E+00	0.	0.	0.	1.0000E+00	0.	0.	0.	1.0000E+00	0.	0.	0.	1.0000E+00	0.	0.	0.	1.0000E+00	0.	0.	0.																				
3	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.																				

Table 4.17: The ME Matrix

THE ME		MATRIX (21 BY 8)			
		1	2	3	4
1	6.7398E-05	-4.6154E-03	-2.5821E-04	-4.9274E-03	
2	-1.5203E-03	9.3534E-04	-5.1396E-03	-7.8871E-04	
3	-1.5071E-03	7.9167E-04	-5.1753E-03	1.4790E-03	
4	1.9461E-03	-7.8443E-03	2.3592E-06	1.0836E-06	
5	-6.6280E-05	4.6117E-03	-2.5814E-04	-4.9299E-03	
6	1.5206E-03	-9.3542E-04	-5.1395E-03	-7.8891E-04	
7	1.5068E-03	-7.9106E-04	-5.1753E-03	1.4799E-03	
8	-6.0352E-03	-1.5745E-03	-3.4857E-08	-4.4647E-09	
9	8.6611E-05	-4.9438E-03	-3.2884E-04	-3.8926E-04	
10	-1.5172E-03	9.1748E-04	-5.1520E-03	-3.3665E-04	
11	-1.5096E-03	8.0491E-04	-5.1720E-03	1.0263E-03	
12	-8.6955E-05	4.9415E-03	-3.2895E-04	-3.8960E-04	
13	1.5172E-03	-9.1803E-04	-5.1520E-03	-3.3656E-04	
14	1.5096E-03	-8.0496E-04	-5.1720E-03	1.0270E-03	
15	-1.5215E-03	1.0421E-03	-5.1323E-03	-1.9280E-03	
16	1.5219E-03	-1.0431E-03	-5.1322E-03	-1.9286E-03	
17	-6.3131E-03	1.4522E-02	2.5935E-07	-3.9864E-07	
18	8.5420E-05	-4.9453E-03	-3.2895E-04	-3.8915E-04	
19	-1.5036E-03	6.9308E-04	-5.1946E-03	2.6169E-03	
20	-8.5364E-05	4.9448E-03	-3.2902E-04	-3.8942E-04	
21	1.5030E-03	-6.9042E-04	-5.1947E-03	2.6184E-03	
		5	6	7	8
1	2.7860E-04	1.5566E-02	-3.7921E-04	-6.0651E-05	
2	7.7980E-03	2.6261E-03	-3.4506E-03	8.1133E-05	
3	7.8853E-03	-4.5157E-03	-3.5444E-03	1.4746E-04	
4	-7.1741E-06	-6.1490E-05	1.7048E-02	4.8504E-04	
5	2.7837E-04	1.5583E-02	4.8135E-04	6.0648E-05	
6	7.7978E-03	2.6072E-03	3.4683E-03	-8.1247E-05	
7	7.8853E-03	-4.5461E-03	3.5145E-03	-1.4714E-04	
8	1.0398E-07	2.3312E-06	-4.7891E-04	-3.0858E-04	
9	4.4976E-04	1.2568E-03	-5.9158E-04	6.4021E-05	
10	7.8314E-03	1.2122E-03	-3.4728E-03	9.5121E-05	
11	7.8791E-03	-3.0951E-03	-3.5398E-03	1.4169E-04	
12	4.5023E-04	1.2500E-03	5.9857E-04	-6.4624E-05	
13	7.8314E-03	1.1903E-03	3.4807E-03	-9.5518E-05	
14	7.8791E-03	-3.1239E-03	3.5190E-03	-1.4084E-04	
15	7.7864E-03	6.2495E-03	-3.4045E-03	6.0750E-05	
16	7.7858E-03	6.2355E-03	3.4456E-03	-6.1232E-05	
17	-5.7021E-07	-1.0374E-05	1.4729E-03	-5.0878E-04	
18	4.5009E-04	1.2558E-03	-5.9276E-04	5.9931E-05	
19	7.9327E-03	-8.1253E-03	-3.6105E-03	1.8241E-04	
20	4.5043E-04	1.2482E-03	6.0083E-04	-5.9368E-05	
21	7.9330E-03	-8.1624E-03	3.5576E-03	-1.8238E-04	

Table 4.18: The MR Matrix

THE MR		MATRIX (21 BY 2)	
	1	2	
1	5.6320E+00	-4.5449E-07	
2	4.4626E+00	4.0000E+00	
3	-5.5374E+00	4.0000E+00	
4	0.	-5.6320E+00	
5	5.6320E+00	4.5449E-07	
6	4.4626E+00	-4.0000E+00	
7	-5.5374E+00	-4.0000E+00	
8	0.	1.4368E+01	
9	-1.4368E+01	-4.5449E-07	
10	2.4626E+00	4.0000E+00	
11	-3.5374E+00	4.0000E+00	
12	-1.4368E+01	4.5449E-07	
13	2.4626E+00	-4.0000E+00	
14	-3.5374E+00	-4.0000E+00	
15	9.4626E+00	4.0000E+00	
16	9.4626E+00	-4.0000E+00	
17	0.	1.4368E+01	
18	-1.4368E+01	-4.5449E-07	
19	-1.0537E+01	4.0000E+00	
20	-1.4368E+01	4.5449E-07	
21	-1.0537E+01	-4.0000E+00	

4.4.2 The Telescope SASLQG Problem

The telescope model defined by (4.27)-(4.33) and Tables 4.15-4.17 will be labeled S_{tele} (24,3,21,45). The following SASLQG problem is posed for the solar optical telescope:

Telescope SASLQG Problem (T.SAS Problem)

Given: S_{tele} (24,3,21,45) with only 12 actuators and 12 sensors available for designing an LQG regulator to achieve the specifications of Table 4.14.

Required: Specify the closed-loop system which satisfies either the input constrained specifications of (3.20) or the output constrained specifications of (3.21).

The T.SAS problem coupled with the H.SAS problem will be the two examples used to test the design algorithm of Chapter 7. With the models of the hoop-column and solar optical telescope developed the discussion of the design algorithm begins in Chapter 5 with the development of the sensor and actuator effectiveness values.

5.0 ACTUATOR AND SENSOR EFFECTIVENESS VALUES

As noted in the introduction, the approach taken by this research to solve the SAS problem has been to use an LQG controller with a cost functional (V of (3.11)) tailored to the specifications (σ^2 , μ^2) of (1.8) by an appropriate choice of the weighting matrices Q and R in V . Then, actuator and sensor deletion decisions are based on a determination of how effective each actuator and sensor is in minimizing V . Those that are least effective are natural candidates for deletion. Chapter 6 discusses the Q and R weight selection problem while the problem of determining actuator and sensor effectiveness values for a specified V is discussed in this chapter. Section 5.1 defines the fundamental elements of these values which are produced by cost analysis techniques, and Section 5.2 combines these elements into the desired sensors and actuator effectiveness values and provides empirical support for their validity.

5.1 Closed-loop Input and Output Cost Analysis

The first step in determining the actuator and sensor effectiveness values is to determine the contribution that each actuator (u_i), actuator noise source (w_i), and sensor noise source (v_i) is making to the minimization of V . The contribution that each output (y_i) makes to V is not necessary for the development of sensor and actuator effectiveness values; however, it is certainly of interest and is directly

related to $E_{\omega} y_i^2$ as will be shown shortly.

Also, it is not necessary to determine the contribution of noise sources (w_i^0) associated with D_0 in (1.1) since these noise sources represent disturbances and/or model errors which are assumed to be independent of the actuator noise sources. That is, the definition of w in (1.1) can be further extended to:

$$(5.1) \quad w \triangleq \begin{pmatrix} w^a \\ w^0 \end{pmatrix} ; \quad E(w(t)w^T(\tau)) \triangleq \begin{bmatrix} w^a & 0 \\ 0 & w^0 \end{bmatrix} \delta(t-\tau)$$

where w_i^a represents the plant noise source associated with the i^{th} actuator.

5.1.1 Closed-loop Cost Definitions

Shown in (5.2) is a partial reproduction of the closed-loop representation (3.3) for $S(n,k,m,l)$ and the LQG regulator of (3.2)

$$(5.2) \quad \begin{aligned} \dot{x} &\triangleq Ax + Bw ; & x &\triangleq (x^T \hat{x}^T)^T ; & w &= (w^T v^T)^T \\ y &= Cx ; & y &= (y^T u^T)^T \\ v &= E_{\omega} y^T Q y ; & Q &= \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix} \end{aligned}$$

The inputs to (5.2) are the white noise processes w and v , and therefore, the technique of determining the contributions that w_i and v_i make to v has been labeled by Skelton and co-workers as closed-loop input-cost analysis (CICA). As a matter of notation, the contribution that w_i makes to v is labeled v_i^w while the contribution of v_i is labeled v_i^v . In like fashion, since y and u appear in the output of the closed-loop

system (5.2), the determination of the contributions that y_i and u_i make to V is called closed-loop output cost analysis (COCA) and the contributions are labeled V_i^y and V_i^u respectively. The mathematical definitions for V_i^u , V_i^y , V_i^w and V_i^v are:

$$\begin{aligned}
 V_i^u &\triangleq \frac{1}{2} E_{\infty} \left\{ \frac{\partial y^T Q y}{\partial u_i} u_i \right\} \\
 V_i^y &\triangleq \frac{1}{2} E_{\infty} \left\{ \frac{\partial y^T Q y}{\partial y_i} y_i \right\} \\
 V_i^w &\triangleq \frac{1}{2} E_{\infty} \left\{ \frac{\partial y^T Q y}{\partial w_i} w_i \right\} \\
 V_i^v &\triangleq \frac{1}{2} E_{\infty} \left\{ \frac{\partial y^T Q y}{\partial v_i} v_i \right\}
 \end{aligned}
 \tag{5.3}$$

5.1.2 Closed-loop Cost Formula's and Properties

The following theorem produces formula's for the definitions of (5.3).

Theorem 1: CICA and COCA Formulas

Given a system of type $S(n,k,m,l)$ regulated by the steady state LQG controller of (3.2) and written in the form of (3.3) the following formulas for V_i^u , V_i^y , V_i^w , and V_i^v hold:

$$(5.4a) \quad V_i^u = [G \hat{X} G^T R]_{ii}, \quad i = 1, \dots, m$$

$$(5.4b) \quad V_i^y = [C(P + \hat{X})C^T Q]_{ii}, \quad i = 1, \dots, k$$

$$(5.4c) \quad v_i^w = \begin{cases} v_i^{w^a} & \forall i = 1, \dots, m \\ v_i^{w^o} & \forall i = m+1 \dots p \end{cases}$$

where,

$$(5.4d) \quad v_i^{w^a} = [B^T(K+L)BW^a]_{ii}, \quad i = 1, \dots, m$$

and

$$(5.4e) \quad v_i^{w^o} = [D_o^T(K+L)D_o W^o]_{ii}, \quad i = 1, \dots, (p-m)$$

$$(5.4f) \quad v_i^v = [F^T L F V]_{ii} \quad i = 1, \dots, \ell$$

where, the matrix L satisfies the following steady state Lyapunov equation

$$(5.4g) \quad L(A-FM) + (A-FM)^T L + G^T R G = 0$$

The proof of theorem 1 is presented in Appendix A, and the formulas of theorem 1 can be used to establish the following properties for v_i^u , v_i^y , v_i^w and v_i^v .

Property 1: Cost Decomposition

$$(5.5a) \quad v = \sum_{i=1}^m v_i^u + \sum_{i=1}^k v_i^y$$

$$(5.5b) \quad v = \sum_{i=1}^p v_i^w + \sum_{i=1}^{\ell} v_i^v$$

Property 2: Sign (For diagonal R , Q , W , and V matrices)

(a) $v_i^u \geq 0$, $v_i^y \geq 0$, $v_i^w \geq 0$, $v_i^v \geq 0$. In addition,

(b) (A,C) observable $\Rightarrow v_i^w > 0$

(c) (A,C) observable, (A+BG,F) controllable $\Rightarrow v_1^u, v_1^y > 0$

(d) (A,D) controllable $\Rightarrow v_1^y > 0$

(e) (A,D) controllable, (A-FM,G) observable $\Rightarrow v_1^w, v_1^v > 0$

Property 3: Transformation invariance

v_1^u, v_1^y, v_1^w and v_1^v are invariant under the state transformation $x = Tq, |T| \neq 0$

Property 4: In situ nature

v_1^u, v_1^y, v_1^w and v_1^v do not, in general, represent the change in V when an actuator, output, noise source, or sensor is deleted from the system. (i.e. the cost contributions are non-linear functions of input, output and sensing terms and are calculated under the assumption that all components are in place and acting).

The proof of properties 1-3 are presented in Appendix A. One other point of interest for these cost contributions is the relationship between v_1^u and $E_{uu_1}^2$ and v_1^y and $E_{yy_1}^2$ when Q and R are assumed diagonal. Writing out the ii element of the matrices in (5.4a,b) and comparing the results with (3.19) and (3.16) produces the following results:

$$(5.6) \quad E_{uu_1}^2 = v_1^u r_1^{-1}$$

$$(5.7) \quad E_{yy_1}^2 = v_1^y q_1^{-1}$$

5.2 Derivation of Effectiveness Values

With the preceding definitions for v_1^u, v_1^w and v_1^v the actuator and sensor effectiveness values can now be derived. The actuator value will be discussed first.

5.2.1 Actuator Effectiveness Value (v_i^{act})

As noted in section 5.1, v_i^u represents the contribution that u_i is making to V . Since the function of LQG theory is to use u_i so that V (for a given Q and R) is minimized, a 'large' v_i^u means that the u_i is *important* to the minimization effort. Furthermore, $v_i^u > v_j^u$ implies that the i^{th} control is more important than the j^{th} control in minimizing V for the given $S(n,k,m,l)$ and Q and R . On the other hand, $v_i^{w^a}$ represents the contribution that the i^{th} actuator noise source is making to V which is clearly an undesirable result. Therefore, $v_i^{w^a} > v_j^{w^a}$ implies that the j^{th} actuator is more important to the minimization effort than the i^{th} actuator. (i.e. it hinders less)

In light of the preceding discussion some combination of v_i^u and $v_i^{w^a}$ could be used to form an actuator effectiveness value. The two most obvious combinations are $v_i^u/v_i^{w^a}$, $v_i^u - v_i^{w^a}$, and these along with some not so obvious combinations have been tested during the course of this research. With the exception of a constrained input power situation which will be discussed in Chapter 7, $v_i^u - v_i^{w^a}$ has been the best combination. It is also intuitively appealing since a linear combination maintains the contributive nature of the values v_i^u and $v_i^{w^a}$. Therefore,

$$(5.8) \quad v_i^{\text{act}} \triangleq v_i^u - v_i^{w^a} ,$$

where v_i^{act} represents the effectiveness value of the i^{th} actuator. A negative value of v_i^{act} means that the i^{th} actuator is contributing more noise than control action to the minimization of V and is therefore a candidate for improved noise characteristics, but, more importantly,

a negative value for V_i^{act} implies that the regulator might do better (i.e. achieve a smaller V) if the i^{th} actuator were deleted! This condition is impossible for noiseless actuators as shown by [theorem 1, 15]. However, for noisy actuators, this condition has been verified by data in [16] and is supported by the following theorem which is proved in Appendix A.

Theorem 2: Deletion of Noisy Actuators

For a system of type $S(n,k,m,\ell)$ under the regulation of the LQG controller defined in (3.2), deletion of an actuator is *not* sufficient for $V(m-1,\ell) > V(m,\ell)$ where

(5.9a) $V(m,\ell) \triangleq$ The value of V for a system of type $S(n,k,m,\ell)$ under LQG regulation.

(5.9b) $V(m-1,\ell) \triangleq$ the value of V for a system of type $S(n,k,m-1,\ell)$ under LQG regulation.

Given theorem 2, the following definition is important.

Definition 1: ΔV_i^{act}

(5.10) $\Delta V_i^{\text{act}} \triangleq V(m-1,\ell) - V(m,\ell)$ where the i^{th} actuator has been deleted in $V(m-1,\ell)$

Therefore if ΔV_i^{act} is negative, the LQG controller does better with a fewer number of actuators, and a positive ΔV_i^{act} implies the opposite.

At this point, it should be remembered that the role of V_i^{act} in the solution of the SASLQG problem is to identify the actuator(s) which are least important to the minimization of V . The preceding discussion

strongly alludes to the fact that the actuators with the smallest *algebraic* values of v_i^{act} are the candidates for deletion. The validity of this argument rests squarely on the assumption of the following ordering property:

$$(5.11) \quad v_i^{\text{act}} \geq v_j^{\text{act}} \Rightarrow \Delta v_i^{\text{act}} \geq \Delta v_j^{\text{act}}$$

In other words, if $v_i^{\text{act}} \geq v_j^{\text{act}}$ throwing away the i^{th} actuator will produce a less favorable perturbation in v than throwing away the j^{th} actuator. Unfortunately, a proof for (5.11) has not been found; however intuition and empirical results point strongly to its validity. Figures 5.1-5.3 are examples of data which provide empirical support for (5.11). Figure 5.1 is a plot of v_i^{act} for the system $S_{\text{Hoop}}(26, 24, 12, 39)$ superimposed with Δv_i^{act} (i.e. $v(11,39) - v(12,39)$ for each of the 12 actuators. The Q and R matrices used in v are given in table 5.1.

Table 5.1: Hoop Column Weighting Matrices, 1

$$Q = \text{diag} [82.07, 82.07, .8207, 82.07, 82.07, .8207, \\ 400,000, \dots, 400,000] \times 10^5$$

18 entries

$$R = 10^{-5} \times I_{12}$$

It should also be noted for figure 5.1 that the actuator number, as defined in table 4.3, is plotted on the horizontal scale in order of decreasing v_i^{act} .

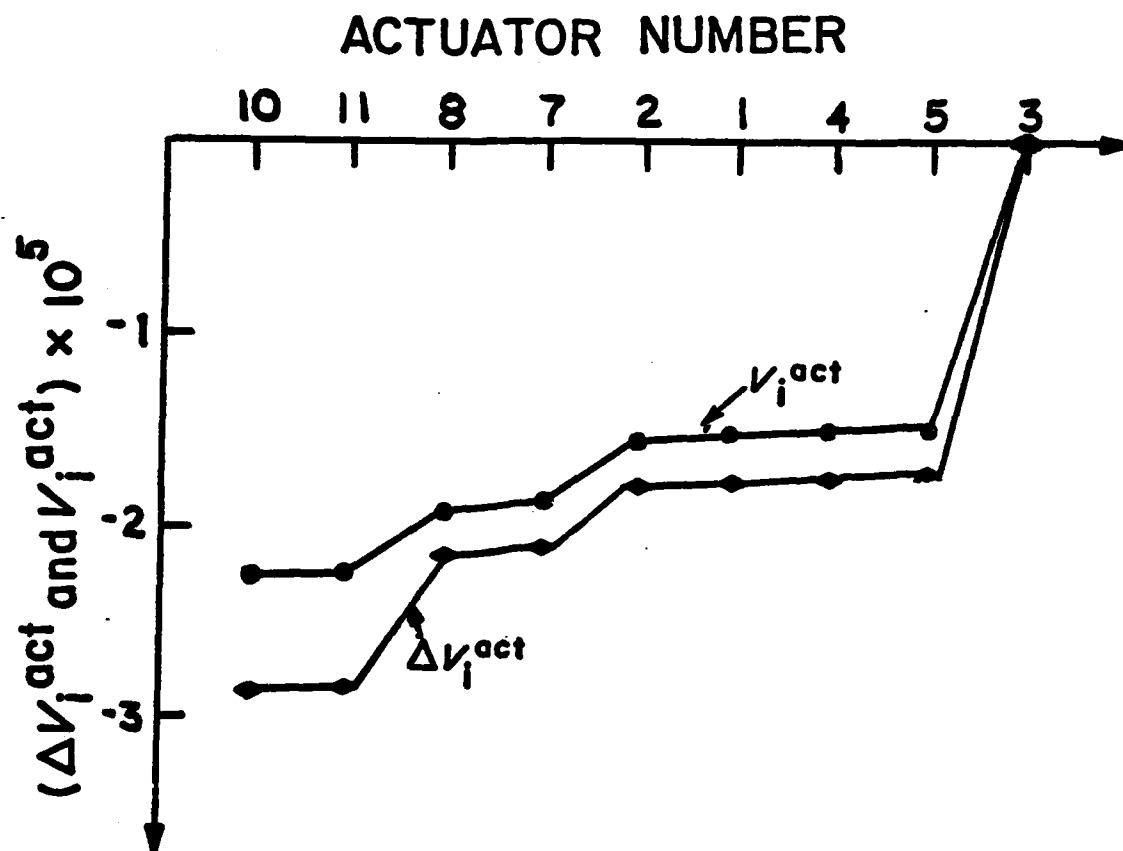


Figure 5.1: Hoop Column v_i^{act} Data for Q and R of Table 5.1

Since the actuators are plotted on the horizontal scale in order of decreasing v_i^{act} , the monotonic decrease of the data in figure 5.1 is empirical support for the ordering property of (5.11). Note also that v_i^{act} does have negative values which supports theorem 2 and shows [theorem 1, 15] does not apply in the noisy actuator situation defined for $S(n, k, m, \epsilon)$. Figure 5.2 is another plot of actuator effectiveness data for $S_{\text{Hoop}}(26, 24, 12, 39)$ with the choice of Q and R defined in table 5.2.

AD-A135 442

SELECTION OF NOISY SENSORS AND ACTUATORS FOR REGULATION 2/3
OF LINEAR SYSTEMS(U) AIR FORCE INST OF TECH

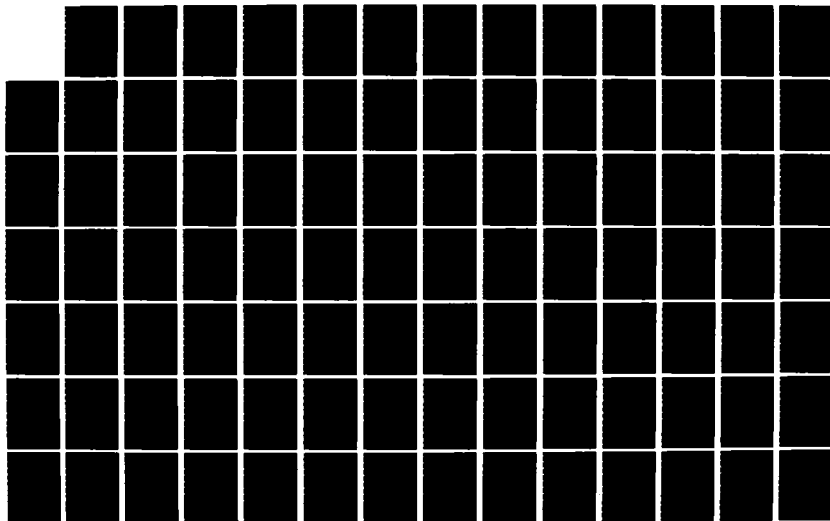
WRIGHT-PATTERSON AFB OH M L DELORENZO AUG 83

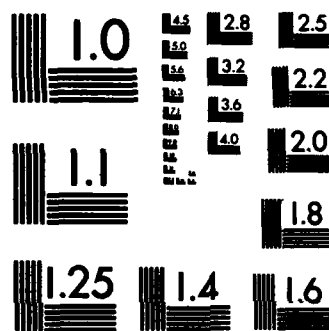
UNCLASSIFIED

AFIT/CI/NR-83-59D

F/G 12/1

NL



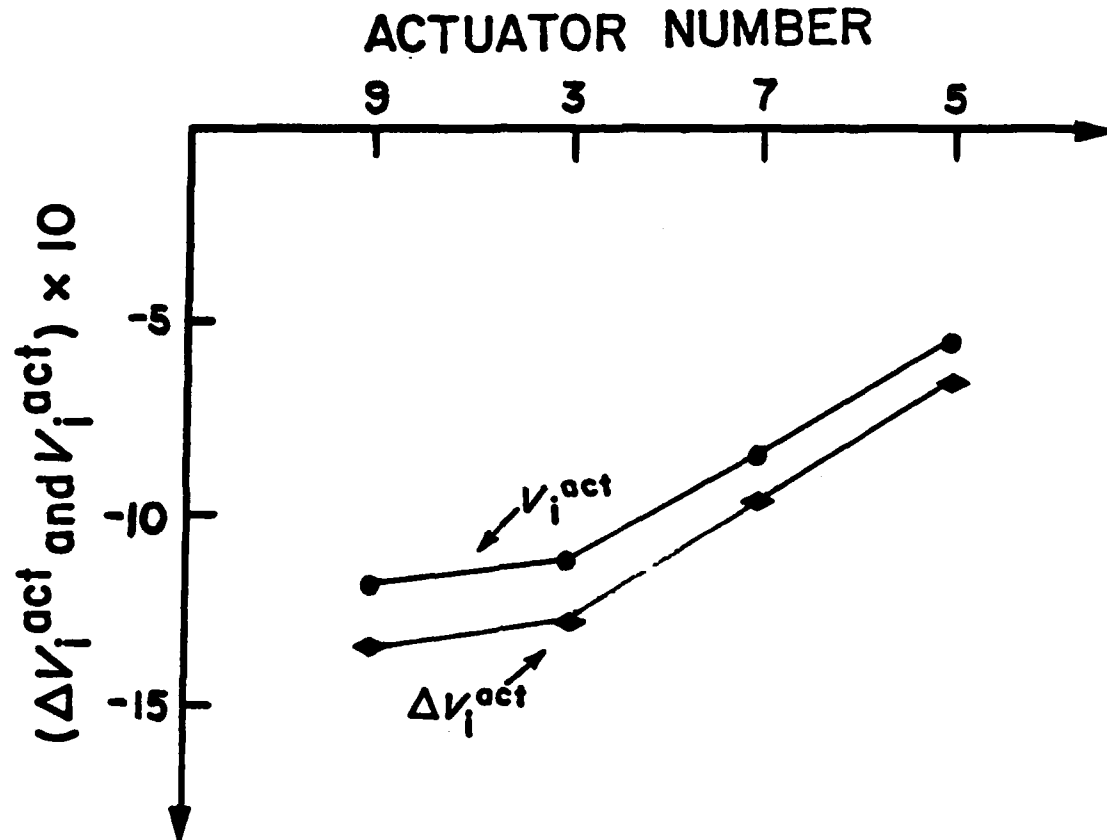


MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Table 5.2: Hoop Column Weighting Matrices, 2

$$Q = \text{diag} \left[\underbrace{0.8207, \dots, 0.8207}_{6 \text{ entries}}, \underbrace{0.4, \dots, 0.4}_{18 \text{ entries}} \right] \times 10^8$$

$$R = 10^{12} \times I_{12}$$

Figure 5.2: Hoop Column V_i^{act} Data
for Q and R of Table 5.2.

The data of figure 5.2 also exhibits a monotonic decrease which agrees with the ordering property of (5.11). A comparison of figure 5.2 and 5.1 reveals one other interesting fact which will be used later in Chapter 7: The relative V_i^{act} ranking between actuator 3, 7, and 5

changed as a function of Q and R . That is the ranking (3,5,7) in figure 5.1 became (5,7,3) in figure 5.2.

Actuator effectiveness data was also obtained for the solar optical telescope (i.e. $S_{\text{Tele}}(24,3,21,45)$). A sampling of this data is shown in figure 5.3, and the weighting matrices are defined in table 5.3.

Table 5.3: Telescope Weighting Matrices

$$Q = \text{diag} [10^8, 10^8, 10^{13}]$$

$$R = 10^5 \times I_{21}$$

The data of figure 5.3 is presented in the same format as figures 5.1 and 5.2 where the actuator numbers are now defined by table 4.13 and $\Delta V_1^{\text{act}} = V(20,45) - V(21,45)$.

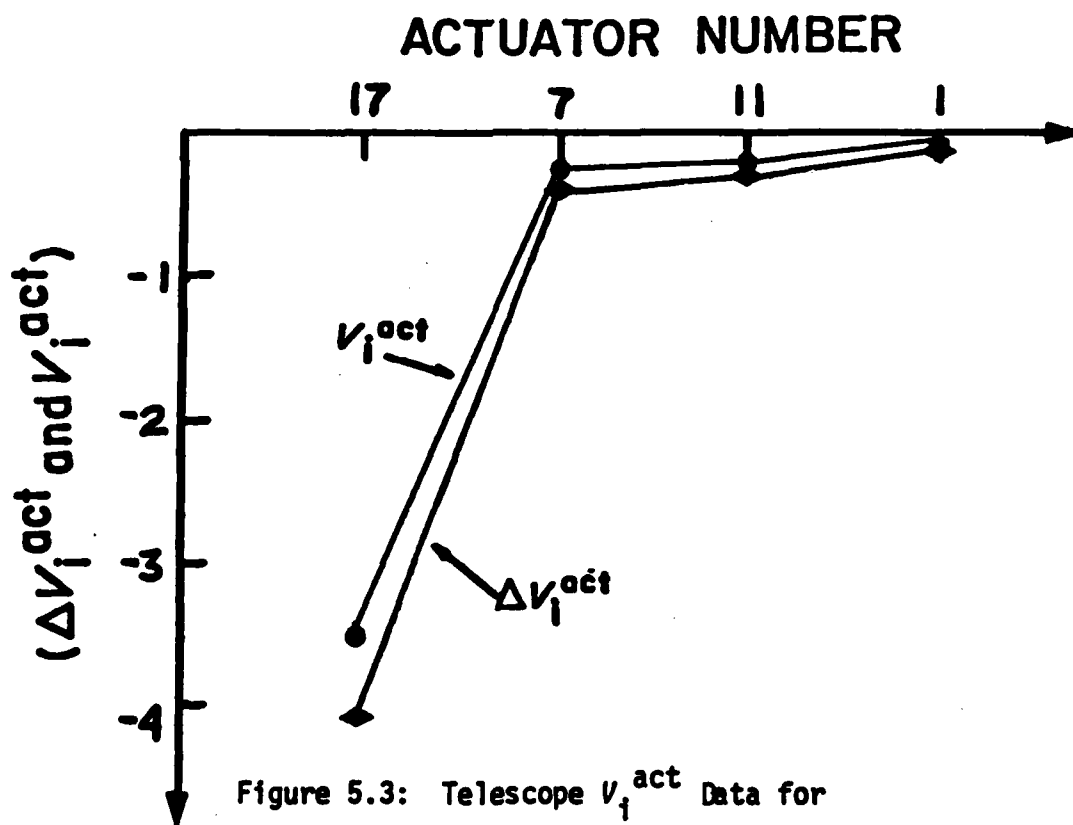


Figure 5.3: Telescope V_1^{act} Data for Q and R of Table 5.3.

Again, the monotonic decreasing nature of the data is empirical support for (5.11).

In summary, all the pertinent data gathered in this research excluding the one special case of constrained input power discussed in Chapter 7, has supported the choice of (5.8) for the actuator effectiveness value to be used in the SASLQG algorithm. Furthermore, as evidenced by figures 5.1-5.3, v_i^{act} appears to be a 'good' estimate of both the sign and magnitude of Δv_i^{act} .

The two major concerns for (5.8) are the lack of a proof for the ordering property of (5.11), and the inability of (5.8) to account for the possibility of the loss of controllability or stabilizability of the system if a particular actuator is deleted. The second concern is further addressed in Chapters 7 and 9.

5.2.2 Sensor Effectiveness Value (v_i^{sen})

It is again worth noting that the role of the actuator and sensor effectiveness values in the SASLQG Algorithm is to identify those sensors and actuators which are contributing the *least* to the minimization of V where V has been chosen through Q and R selection to insure that the LQG controller achieves desired input and output variances.

As discussed in Section 5.1, v_i^V represents the contribution that the i^{th} sensor noise source makes to V . Therefore, those actuators with larger values for v_i^V are contributing more noise to V than those actuators with lesser v_i^V , and at first glance appear to be candidates for deletion. However, the sensor measurements for $S(n,k,m,\ell)$ under LQG regulation are being passed through a Kalman-

Bucy filter whose function is to "de-emphasize" or "throw-out" measurements which have more noise than estimation information.[1] Therefore, any noise source that is making a 'large' contribution to V emanates from a sensor which is making an even 'larger' contribution to the estimation information necessary to minimize V . This is further supported by taking an expanded look at the formula for V_i^V in (5.4f). Assuming a diagonal V :

$$(5.12) \quad V_i^V = m_i^T P L P m_i V_{ii}^{-1} \quad \text{where } m_i \text{ is the } i^{\text{th}} \text{ col of } M^T$$

Equation (5.12) shows that V_i^V is an explicit as well as an implicit function of V_{ii}^{-1} which is the inverse of the variance of the i^{th} sensor noise source. Therefore, V_i^V will tend to be *larger* for sensors with *smaller* noise variances! Of course, the estimation information contained in $m_i^T P L P m_i$ will have a significant effect. Also, in Appendix B, a development is presented which shows a significant correlation between V_i^V and the Chen-Seinfeld Switching functions. Chiu, in [15], has shown that sensors with *larger* values for those switching functions are the ones which satisfy the necessary conditions for minimizing V .

In light of the above discussion an obvious choice for the sensor effectiveness value is:

$$(5.13) \quad V_i^{\text{sen}} \triangleq V_i^V$$

Sensors with the smallest values for V_i^V are then taken as candidates for deletion. The following definition, which is analogous to

Definition 1 in Section 5.2.1 is now germane to the discussion.

Definition 2: Δv_i^{sen}

$$(5.14) \quad \Delta v_i^{\text{sen}} \triangleq v(m, l-1) - v(m, l)$$

where $v(m, l)$ is defined in (5.9a) and $v(m, l-1)$ is the sensor dual of (5.9b)

As in the case of v_i^{act} the essential property that must be satisfied by v_i^{sen} is the ordering property:

$$(5.15) \quad v_i^{\text{sen}} \geq v_j^{\text{sen}} \rightarrow \Delta v_i^{\text{sen}} \geq \Delta v_j^{\text{sen}}$$

Again as in the case of v_i^{act} , only the intuitive arguments of the preceding discussion and empirical results currently exist to verify the ordering property of (5.15).

Figures 5.4-5.6 contain data, which support the validity of (5.15). Figure 5.4 is a plot of v_i^{sen} for the system $S_{\text{Hoop}}(26, 24, 12, 39)$ superimposed with Δv_i^{sen} (i.e. $v(12, 38) - v(12, 39)$) for seven sensors. The Q and R matrices used are those defined by table 5.1. It should further be noted that sensor numbers on the horizontal scale are defined by table 4.2, and they are ordered from left to right in terms of decreasing v_i^{sen} . The monotonic decrease of the data in figure 5.4 is empirical support for the ordering property of (5.15). Figures 5.5 and 5.6 show data analogous to figure 5.4 for $S_{\text{Hoop}}(26, 24, 12, 39)$ with the weights of table 5.2 and $S_{\text{tele}}(24, 3, 21, 45)$ with the weights of table 5.3 respectively.

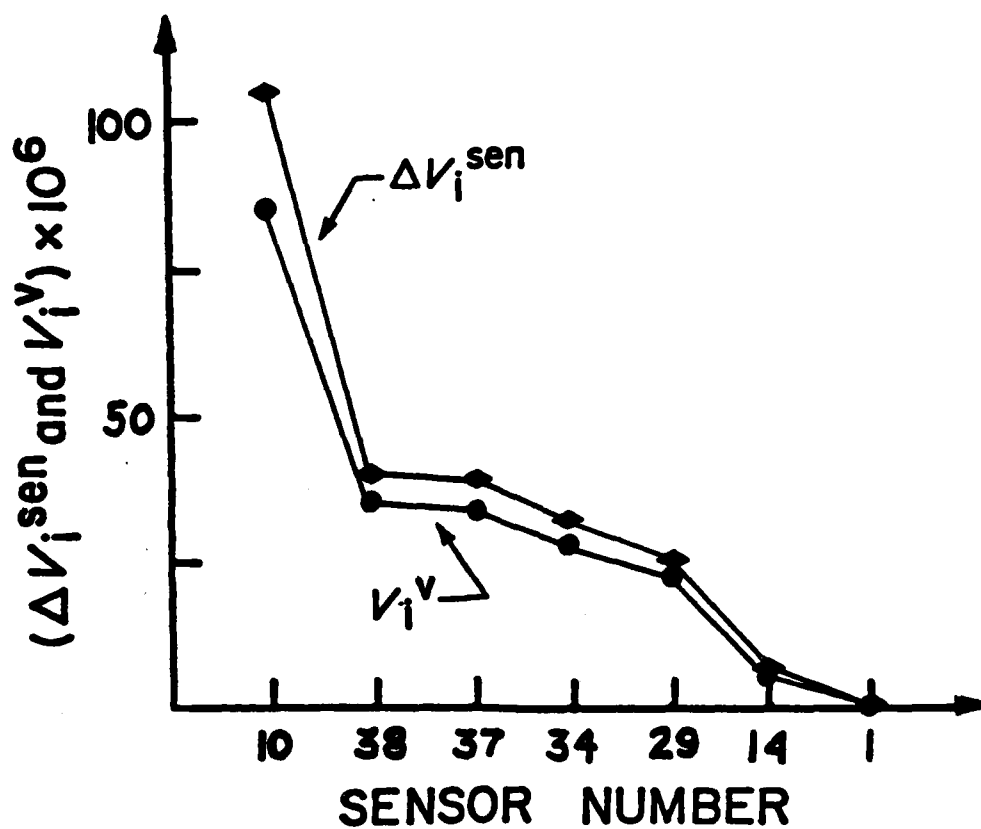


Figure 5.4: Hoop Column V_i^{sen} Data
for Q and R of Table 5.1.

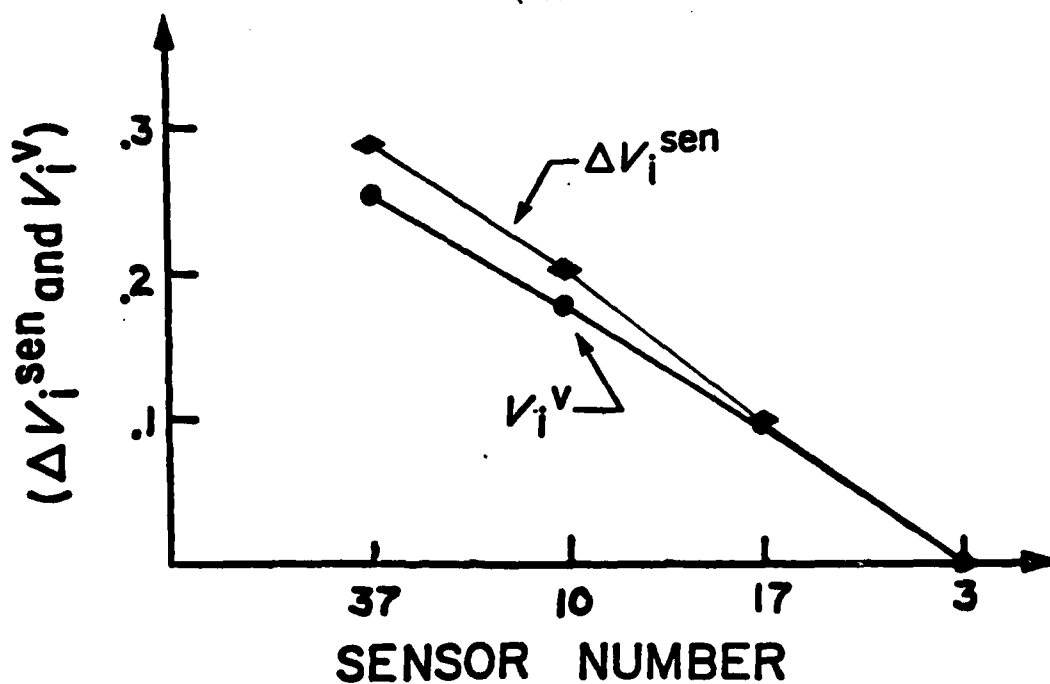


Figure 5.5: Hoop Column V_i^{sen} Data for
Q and R of Table 5.2.

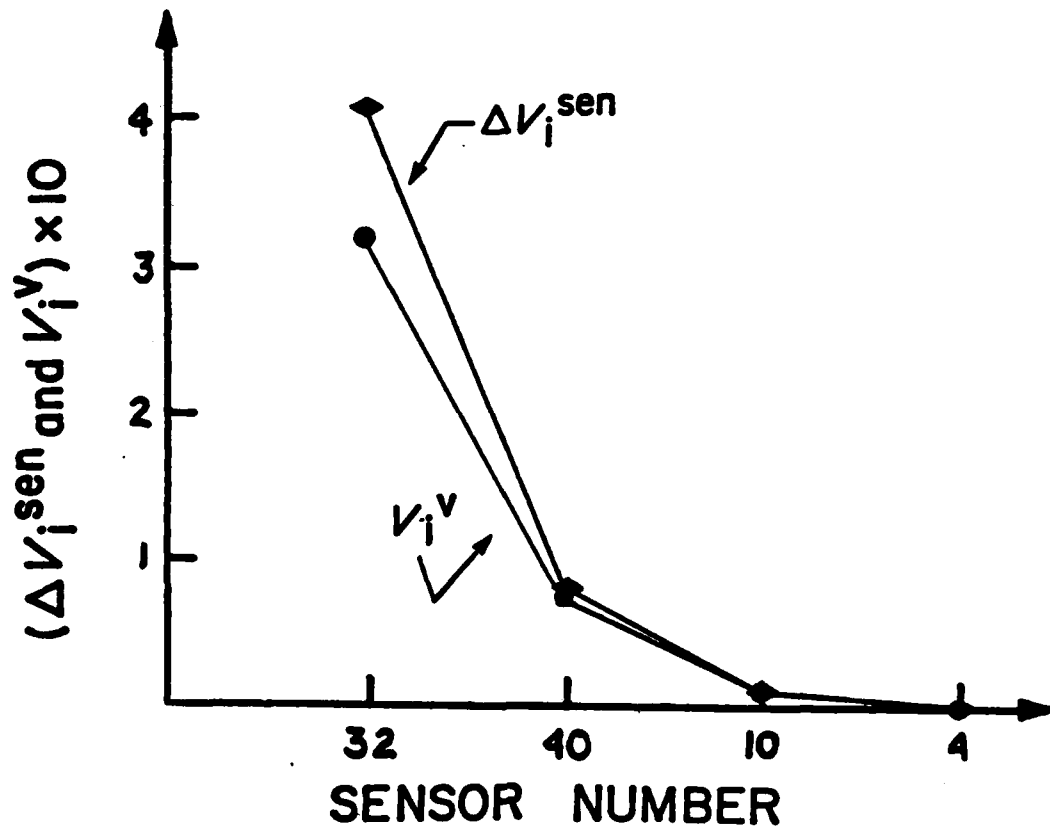


Figure 5.6: Telescope V_i^{sen} Data for Q and R of Table 5.3.

The data of both figures again exhibit a monotonic decrease and offer further support for (5.15).

Another point of interest surfaces when comparing figures 5.4 and 5.5. This comparison reveals that the relative V_i^{sen} ranking of sensors 10 and 37 changed as a function of Q and R, a matter of importance to the development of Chapter 7. Also, in none of the data of figures 5.4-5.7 did deleting a sensor provide better regulator performance (i.e. V_i^{sen} was never negative). This result is empirical evidence of the following theorem which is proved in [15] and [16] and also in Appendix A.

Theorem 3: Deletion of Noisy Sensors

For a system of type $S(n,k,m,\ell)$ under the regulation of the LQG controller defined in (3.2), deletion of a sensor *cannot* reduce V .

In summary, *all* pertinent data gathered in this research supports the validity of choosing (5.13) to represent the sensor effectiveness value for the SASLQG algorithm. Also, as evidenced by the data of figures 5.4-5.7, v_i^{sen} is a 'good' estimate for both the sign and magnitude of Δv_i^{sen} . However, the same two concerns that exist for v_i^{act} , also exist for v_i^{sen} : There is currently only empirical and intuitive support for the ordering property of (5.15), and v_i^{sen} does not consider that measurability or detectability might be lost when the i^{th} sensor is deleted. The measurability/detectability concern is further discussed in Chapters 7 and 9.

With the expressions for sensor and actuator effectiveness values chosen, the next order of business is to develop an algorithm for selecting Q and R so that the LQG controller which minimizes V also achieves desired variance constraints on system inputs and outputs. This algorithm is the topic of Chapter 6.

6.0 LQG WEIGHT SELECTION

A fundamental step in solving the SASLQG problem is to adjust the elements of Q and R in V such that the resulting LQG controller, for a *fixed* set of sensors and actuators, achieves the input-constrained requirements of (3.20) or the output constrained requirements of (3.21). A statement of this mathematical problem will prove useful. Substituting the definitions for $E_{\omega} y_i^2$ and $E_{\omega} u_i^2$ provided by (3.16) and (3.19) into (3.20) and (3.21) the following statement for a constrained variance LQG (CVLQG) problem results:

CVLQG Problem

Given: A system of type $S(n,k,m,l)$ under the control of an LQG regulator defined by (3.2) and with variance specifications (σ^2, μ^2) of (1.8)

Required: Determine the *diagonal* elements of Q and R such that one of the following holds:

Input Constrained Solution

If the specifications (σ^2, μ^2) are achievable

$$\text{Min}_{Q,R} \sum_{i=1}^k (c_i^T (P + \hat{X}) c_i) / \sigma_i^2$$

(6.1a)

$$\text{subject to } r_i^{-2} b_i^T K \hat{X} K b_i = \mu_i^2 \quad \forall i = 1, \dots, m$$

else,

$$\text{Min}_{Q,R} \sum_{i=1}^k (c_i^T (P+\hat{X}) c_i) / \sigma_i^2 \quad \forall i: c_i^T (P+\hat{X}) c_i > \sigma_i^2$$

(6.1b)

$$\text{subject to } r_i^{-2} b_i^T K \hat{X} K b_i = \mu_i^2 \quad \forall i = 1, \dots, m$$

Output Constrained Solution

If (σ^2, μ^2) are achievable,

$$\text{Min}_{Q,R} \sum_{i=1}^m (r_i^{-2} b_i^T K \hat{X} K b_i) / \mu_i^2$$

(6.2a)

$$\text{subject to } c_i^T (P+\hat{X}) c_i = \sigma_i^2 \quad \forall i = 1, \dots, k$$

else,

$$\text{Min}_{Q,R} \sum_{i=1}^m (r_i^{-2} b_i^T K \hat{X} K b_i) / \mu_i^2 \quad \forall i: r_i^{-2} b_i^T K \hat{X} K b_i > \mu_i^2$$

(6.2b)

$$\text{subject to } c_i^T (P+\hat{X}) c_i = \sigma_i^2 \quad \forall i = 1, \dots, k$$

The CVLQG problem is a non-linear programming problem which has two distinct requirements: The first being to determine if a diagonal Q and R exists to achieve (σ^2, μ^2) , and the other being to adjust the Q and R elements to achieve either the input-constrained or output-constrained solution. An obvious approach to this problem would be to apply standard, non-linear programming techniques. However, these techniques almost always require gradient calculations and a search routine for an appropriate step size. ([46], [59]-[61]) Given the possibility of a large number of inputs and outputs, a large system order and the requirement for solution of an algebraic Riccati equation at each iteration of the algorithm and at each iteration of any

step size search routine, the calculations required by these standard gradient approaches would be prohibitive when applied to the CVLQG problem. This chapter presents an algorithm for solving the CVLQG problem which requires only the standard LQG calculations of (3.2)-(3.6) and hence avoids the computational burdens of gradient calculations and a step size search routine. The algorithm also uses inherent properties of the LQG controller to make update and existence decisions. Before discussing the algorithm, a survey of past techniques and motivations for Q and R adjustments is presented in Section 6.1. Then, Section 6.2 provides a general discussion of input-constrained and output constrained solutions while Section 6.3 presents the important theory behind the algorithm. The specific steps of the algorithm are provided in section 6.4, and the algorithm is applied to the hoop-column antenna model and the telescope model in sections 6.5 and 6.6 respectively.

6.1 Past Approaches to Q and R Selection

Even though LQG theory has a natural application to the problem of satisfying variance constraints on system inputs and outputs, not a great deal has been published on the subject. In [2], Bryson and Ho suggest making Q and R diagonal such that:

$$(6.3) \quad \begin{aligned} q_i &= 1/\sigma_i^2 \\ r_i &= 1/\mu_i^2 \end{aligned} ; \quad \text{where } \sigma_i^2 \text{ and } \mu_i^2 \text{ are components of the specifications } (\sigma^2, \mu^2) \text{ in (1.8)}$$

This choice for Q and R means that V in (3.12) can be written in the following form

$$(6.4) \quad V = \sum_{i=1}^k E_{\infty} y_i^2 / \sigma_i^2 + \sum_{i=1}^m E_{\infty} u_i^2 / \mu_i^2$$

From (6.4) it can be seen that the LQG controller designed by the weighting matrices (6.3) minimizes the sum of the output and input mean square values normalized by their specification. Therefore, selecting (6.4) for Q and R guarantees that the *average* of the normalized mean square values of all components has been minimized. While this is a step in the right direction, it does not guarantee that the LQG controller will meet the requirements of the CVLQG problem. In the past, if using (6.3) did not achieve the desired variance specification it was necessary to resort to trial and error variation of the elements of Q and R based upon a set of general "directional" guidelines. (See [1], [2], [47]).

Recently, in [44], [48] and [49], iterative algorithms have been proposed for adjusting elements of the Q and R matrices in order to achieve desired variance specifications on system inputs and outputs. In [44] the following update equations are proposed.

$$(6.5a) \quad q_i(j+1) = \frac{E y_i^2(j)}{\sigma_i^2} q_i(j) = q_i(j) + \frac{1}{\sigma_i^2} (E_{\infty} y_i^2(j) - \sigma_i^2) q_i(j)$$

$$(6.5b) \quad r_i(j+1) = \frac{E_{\infty} u_i^2(j)}{\mu_i^2} r_i(j) = r_i(j) + \frac{1}{\mu_i^2} (E_{\infty} u_i^2(j) - \mu_i^2) r_i(j)$$

Initially, if *outputs* are out of specification ($E_{\infty} y_i^2 / \sigma_i^2 > 1$), the algorithm in [44] uses only (6.5a) to adjust the weights on those outputs. When all outputs are within specification the algorithm uses (6.5a) on all outputs and (6.5b) only on the inputs that are *within* specification ($E_{\infty} u_i^2 / \mu_i^2 < 1$). The algorithm continues until all specifications are satisfied or until the components that are out of specification are no longer changing.

In [48], the following update equation is used: (assuming Q and R diagonal)

$$(6.6) \quad \lambda(j+1) = \lambda(j) + \beta_j H_j \epsilon(j)$$

where

$$\lambda = [q_1, q_2, \dots, q_k, r_1, \dots, r_m]$$

β_j = scalar step length parameter at iteration j

$$\epsilon(j) = \begin{cases} [E_{\infty} y_i^2 - \sigma_i^2] q_i(j) & i=1, \dots, k \text{ and } E_{\infty} y_i^2 > \sigma_i^2 \\ [E_{\infty} u_i^2 - \mu_i^2] r_i(j) & i=1, \dots, m \text{ and } E_{\infty} u_i^2 > \mu_i^2 \\ 0 & \text{otherwise} \end{cases}$$

H_j = Broyden approximation to the inverse of the Jacobian of $\epsilon(j)$ with respect to $\lambda(j)$, [50].

Therefore, the algorithm of [48] adjusts those components of Q and R which correspond to outputs and inputs that are out of specification. Neither [44] nor [48] give conditions under which the algorithms can

be expected to converge to a solution or specify conditions under which it is known that no choice for Q and R would exist. Also not addressed are the input-constrained and output constrained requirements of (6.1) and (6.2).

In [49], the following update equation for r_i is proposed for each sample time in a self-tuning, discrete time, LQG regulator:

$$(6.7) \quad r_i(k+1) = r_i(k) + \mu(k+1) \frac{r_i(k)}{\mu_i^2} (u^2(k) - \mu_i^2)$$

where $0 < \mu(k) < 1$ and $\mu(k)$ is chosen to insure $r_i(k+1)$ is positive. The objective is to use (6.7) to design a self-tuned LQG regulator which has the inputs operating at their variance specifications μ_i^2 .

Other studies on the selection of weights Q and R have entirely different motivations, but are mentioned here for completeness. Considerable study has been devoted to the selection of Q and R to achieve desired closed-loop pole locations, [51]-[54]. References [55] and [56] relate Q and R selection to such additional frequency domain properties as stability margin and disturbance rejection. In [57] Sesak et al. propose non-diagonal selections for Q and R which serve to suppress LQG controller excitation of unmodeled system states. In work related to [51]-[54] Sesak et al., in [58], propose a technique for selecting non-diagonal Q and R so that the compensator poles (i.e. the eigenvalues of A_c) are within some prespecified stable region.

As a final note, the algorithm presented in this chapter has been documented in [45]. With this background for Q and R presented, the algorithm development begins with a discussion of input and output constrained solutions.

6.2 Input/Output-Constrained Solutions

Before presenting the theory behind the proposed algorithm, a less technical discussion of the input-constrained and output-constrained requirements of (6.1) and (6.2) for single input single output systems (i.e. $S(n,1,1,\ell)$) will prove useful. For $S(n,1,1,\ell)$ the LQG cost functional of (3.12) takes the following form:

$$(6.8) \quad V = E_{\omega} y_1^2 q_1 + E_{\omega} u_1^2 r_1$$

6.2.1 Input-Constrained Discussion ($S(n,1,1,\ell)$)

Assume that r_1 can be adjusted so that $E_{\omega} u_1^2 = \mu_1^2$. Labeling this value for r_1 as r_M (6.8) can be written as:

$$(6.9) \quad V = E_{\omega} y_1^2 q_1 + \mu_1^2 r_M$$

For the given $S(n,1,1,\ell)$, μ_1 , and output weighting q_1 , the term $\mu_1^2 r_M$ represents a fixed control penalty term in V which insures that all available control power (in a stochastic sense) is being used by the LQG controller to minimize $E_{\omega} y_1^2 q_1$. Therefore the term $E_{\omega} y_1^2 q_1$ cannot be made smaller without violating the input constraint. In light of this discussion, the appropriate choice for q_1 to achieve the input-constrained requirement of (6.1) is $1/\sigma_1^2$. Therefore, define $q_1^* = 1/\sigma_1^2$. The preceding situation is represented pictorially in figure 6.1 by making use of the known inverse relationship between control power $E_{\omega} u_1^2$ and output variance $E_{\omega} y_1^2$. [1] The two curves represent the performance lines of the different systems S_a, S_b with output weighting q_1^* . The interior of the rectangular box formed

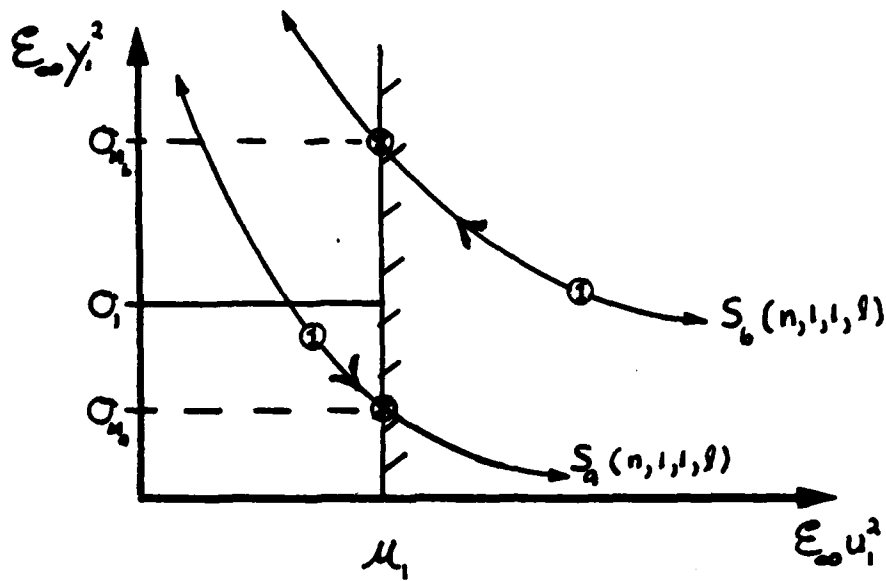


Figure 6.1: $S(n,1,1,l)$ Input-Constrained Solution

by the σ_1^2, u_1 specification lines represents the region in which (σ_1^2, u_1^2) is satisfied. Clearly, an LQG controller does not exist for S_b which can meet (σ_1^2, u_1^2) . Therefore the input-constrained condition (6.1b) is sought. On the other hand, there are essentially an infinite number of LQG controllers which can keep the operating point of S_a within the (σ_1^2, u_1^2) specification region, and the requirement (6.1a) applies. The current operating point of both systems is assumed to be (1) and the objective of the input constrained solution is to move the operating points to (X) by finding r_M . Then σ_{M_b} represents the minimum achievable output specification for (6.1b) and σ_{M_a} represents the minimum achievable output specification for (6.1a).

6.2.2 Output-Constrained Discussion ($S(n,1,1,l)$)

A discussion of the output-constrained solution proceeds in an identical fashion. Assume that a q_1 is found such that $E_{\infty} y_i^2 = \sigma_i^2$. Labeling this value q_M , (6.8) becomes:

$$(6.10) \quad V = \sigma_1^2 q_M^2 + E_{\infty} u_1^2 r_1$$

For the given $S(n,1,1,\ell)$, σ_1 , and input weighting r_1 , the term $\sigma_1^2 q_M^2$ represents a fixed output penalty term in V which produces the maximum allowable output variance. Therefore the term $E_{\infty} u_1^2 r_1$ cannot be reduced without violating the output constraint and $r_1 = 1/\mu_1 \triangleq r_1^*$ satisfies (6.2). Figure (6.2) is a pictorial representation of systems $S_a(n,1,1,\ell)$ and $S_b(n,1,1,\ell)$ with control weighting r_1^* .

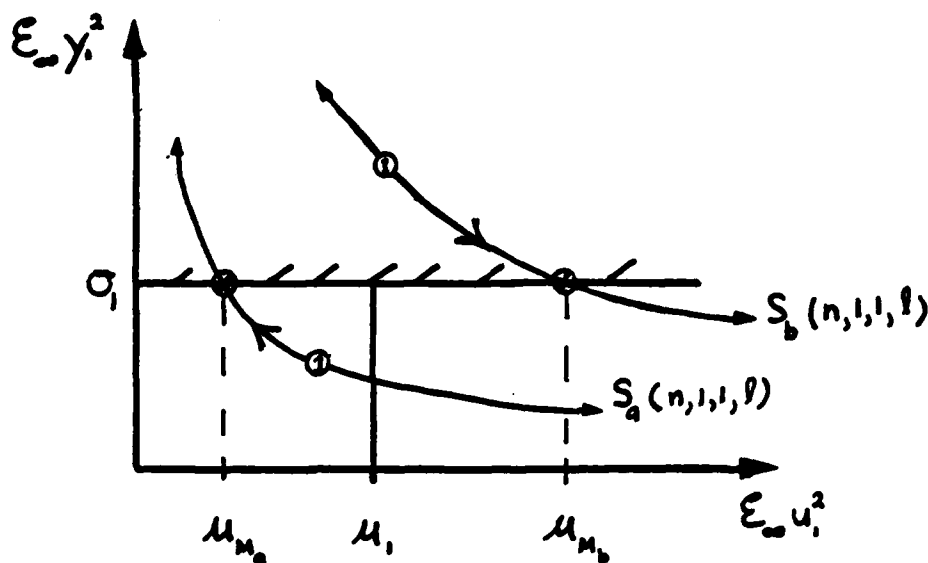


Figure 6.2: $S(n,1,1,\ell)$ Output-Constrained Solution

As in the input constrained case, the (σ_1^2, μ_1^2) specification can be achieved for S_a but not for S_b . The objective of the output-constrained solution is to move the systems operating point from (1) to (X) by finding q_M . Then, μ_{M_a} satisfies (6.2a) and μ_{M_b} satisfies (6.2b).

6.2.3 Multi-Input/Output-Constrained Solutions

There is a subtlety in achieving (6.1) or (6.2) which does not appear in the single-input, single-output case and should also be

discussed before the theory is presented. Assume a single input two output system (i.e. $S(n,2,1,\ell)$) with specifications $(\sigma_1, \sigma_2, \mu_1)$. The pictorial representation of both outputs vs. control effort is shown in Figure 6.3 for an input constrained solution and the output weighting $Q = \text{diag}(q_1^*, q_2^*)$

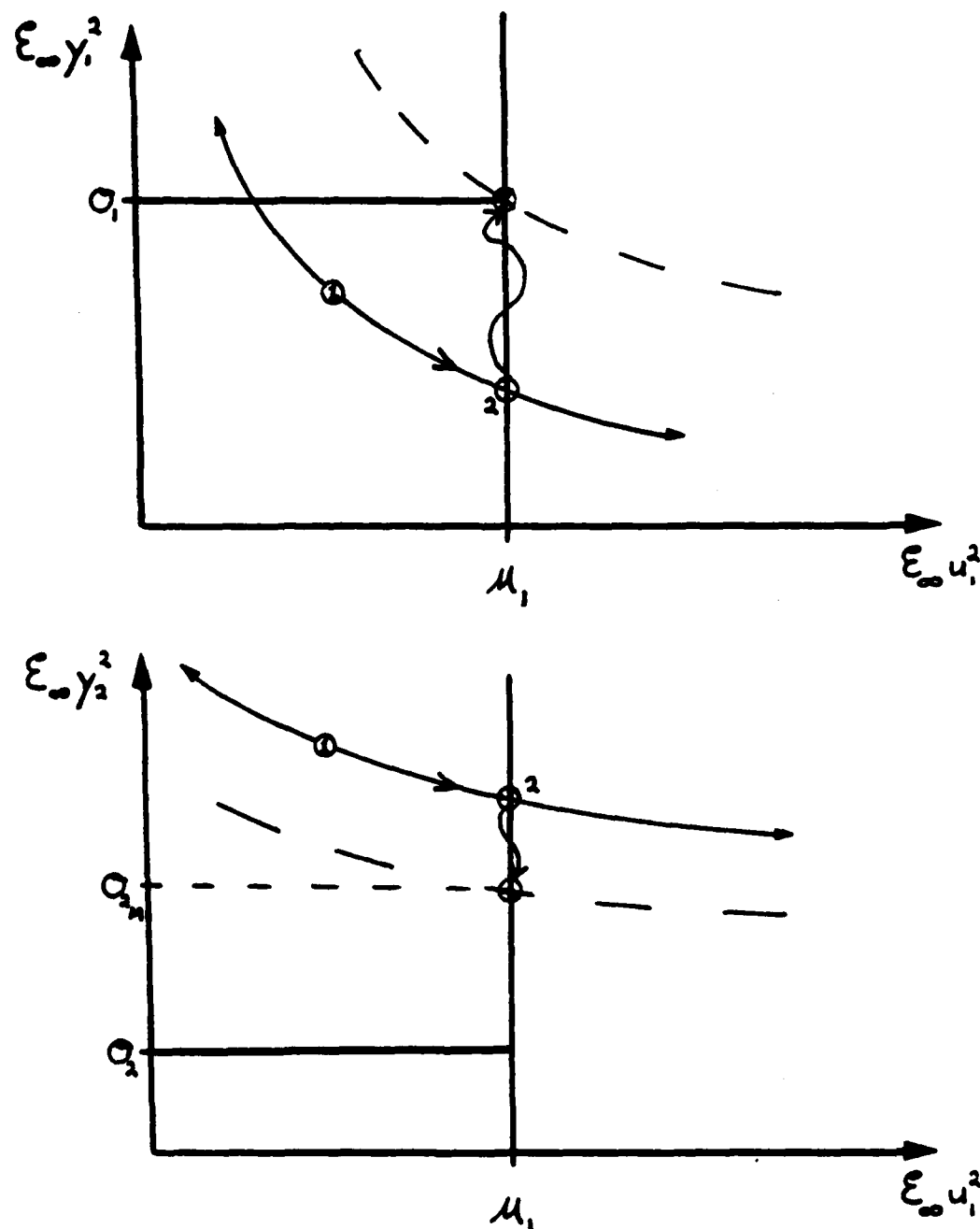


Figure 6.3 $S(n,2,1,\ell)$ Input-Constrained Solution

From the figure it is obvious that σ_2 is restrictive and an LQG controller does not exist for (σ^2, u^2) . Therefore, the input constrained solution adopts the criterion of (6.1b) and moves from (1) to (2) by finding r_M . At point (2) $E_{y_1}^2$ is well within specification and $E_{y_2}^2$ is well above specification. The requirement of (6.1b) in this case is that $E_{y_2}^2$ be as small as possible. Since $E_{y_1}^2$ is well within specification, a logical course of action would be to channel some of the control effort from y_1 to y_2 . For the LQG controller, this is accomplished by decreasing q_1 and adjusting r_M so that all allowable control power is still being used (i.e. keeps the operating point at u_1). This redistribution of control power continues until $E_{y_1}^2$ reaches its specification at point (X) or (although not illustrated in figure 6.3) $E_{y_2}^2$ reaches its specification. Since all available control power has been directed toward the output above specification, (i.e. $E_{y_2}^2$) the requirement of (6.1b) should be satisfied. The specific output weighting condition which achieves the control redistribution pictured in figure 6.3 is $Q = \text{diag}(q_M, q_2^*) = Q_M^*$. An identical situation to the one pictured in figure 6.3 exists for the output-constrained solution for $S(n,1,2,2)$. The input weighting condition for this situation is defined to be R_M^* .

The discussion just presented has emphasized important elements in the solution of the CVLQG problem. It also raises the questions of whether the specific weighting conditions Q^* , R^* , Q_M^* , R_M^* exist and how to find them. The theory presented in the next section addresses these questions.

6.3 Theory

In this section, the supporting theory for the algorithm of Section 6.4 is presented. The following definitions are fundamental to the discussion.

6.3.1 Definitions

The following are formal multiple input/output definitions for the expressions adopted in Section 6.2.

Definition 3: $\nu_{\text{spc}}^y, \nu_{\text{spc}}^u, \nu_{\text{spc}}$

For the system $S(n,k,m,l)$ using the LQG controller of (3.2)-(3.6)

$$(6.11a) \quad \nu_{\text{spc}}^y(Q,R) \triangleq \sum_{i=1}^k E_{\infty} y_i^2 / \sigma_i^2$$

$$(6.11b) \quad \nu_{\text{spc}}^u(Q,R) \triangleq \sum_{i=1}^m E_{\infty} u_i^2 / \mu_i^2$$

$$(6.11c) \quad \nu_{\text{spc}}(Q,R) \triangleq \nu_{\text{spc}}^y(Q,R) + \nu_{\text{spc}}^u(Q,R)$$

where the arguments (Q,R) represent the weighting matrices used in determining the LQG controller.

Definition 4: $Q^*, R^*, Q_M, R_M, Q_M^*, R_M^*$

For the *diagonal* weighting matrices Q, R

$$(6.12a) \quad Q^* = (Q: q_i = 1/\sigma_i^2; i = 1, \dots, k); \sigma_i^2 = \begin{matrix} \text{Variance} \\ \text{Specification} \end{matrix}$$

$$(6.12b) \quad R^* \triangleq (R: r_i = 1/\mu_i^2; i = 1, \dots, m); \mu_i^2 = \text{Variance specification}$$

$$(6.12c) \quad Q_M \triangleq (Q: E_{\omega} y_i^2 = \sigma_i^2 \forall i = 1, \dots, k)$$

$$(6.12d) \quad R_M \triangleq (R: E_{\omega} u_i^2 = \mu_i^2 \forall i = 1, \dots, m)$$

$$(6.12e) \quad Q_M^* \triangleq \left(Q: \begin{cases} E_{\omega} y_i^2 \geq \sigma_i^2 \forall i = 1, \dots, k \\ \text{and} \\ q_i = 1/\sigma_i^2 \forall i: E_{\omega} y_i^2 > \sigma_i^2 \end{cases} \right)$$

$$(6.12f) \quad R_M^* \triangleq \left(R: \begin{cases} E_{\omega} u_i^2 \geq \mu_i^2 \forall i = 1, \dots, m \\ \text{and} \\ r_i = 1/\mu_i^2 \forall i: E_{\omega} u_i^2 > \mu_i^2 \end{cases} \right)$$

It should be noted that Q^* and R^* exist and are unique for any non-zero (σ^2, μ^2) however, Q_M , R_M , Q_M^* , and R_M^* are not unique nor do they always exist. The existence question will be addressed shortly.

6.3.2 Theorems

Using the notation of definitions 3 and 4 the following important theorems are stated and proved. They will be recognized as multi-input/output versions of the situations discussed in Section 6.2.

Theorem 4: Input-constrained conditions

For a system of type $S(n, k, m, \ell)$ under the regulation of the LQG controller defined by (3.2)-(3.6) (diagonal Q and R) and the assumption that the matrices Q_M^* and R_M exist, the following holds:

$$(a) \quad \nu_{\text{spc}}^y(Q^*, R_M) \leq \nu_{\text{spc}}^y(Q, R) \quad \forall \quad Q, R \text{ subject to the constraint that}$$

$$E_{\infty} u_i^2 = \mu_i^2 \quad \forall i = 1, \dots, m$$

$$(b) \quad \nu_{\text{spc}}^y(Q_M^*, R_M) > k \Rightarrow (\sigma^2, \mu^2) \text{ cannot be satisfied by an LQG controller}$$

(c) Given condition (b), the LQG controller designed by (Q_M^*, R_M) minimizes the following:

$$\sum_{i=1}^k E_{\infty} y_i^2 / \sigma_i^2 \quad \forall i: E_{\infty} y_i^2 > \sigma_i^2$$

Proof: From definition 4, the controller defined by (Q^*, R_M) minimizes the cost functional:

$$(6.13) \quad \nu = \underbrace{\sum_{i=1}^k E_{\infty} y_i^2 / \sigma_i^2}_{(1)} + \underbrace{\sum_{i=1}^m \mu_i^2 r_{M_i}}_{(2)}$$

For the given $S(n, k, m, \ell)$ and (σ^2, μ^2) , term (2) is a fixed penalty term in V constrained by the R_M requirement $E_{\infty} u_i^2 = \mu_i^2 \quad \forall i = 1, \dots, m$ (i.e. use all available control power). Under this condition the LQG controller minimizes term (1), and from the known optimality conditions for the LQG controller, no other LQG controller (i.e. different Q, R) or any linear[†] controller, for that matter, can do better for the

[†]If the noise processes in $S(n, k, m, \ell)$ are Gaussian, the LQG controller also does better than any nonlinear controller. [1]

given constraints. [1] Since term (1) is equivalent, by definition, to $\nu_{\text{spc}}^y(Q^*, R_M)$ point (a) of theorem 4 is proved.

Again from definition 4, the controller defined by (Q_M^*, R_M) minimizes the cost functional:

$$(6.14) \quad \nu = \underbrace{\sum_{i=1}^k E_{\omega} y_i^2 / \sigma_i^2}_{(1)} + \underbrace{\sum_{i=1}^k \sigma_i^2 q_{M_i}^*}_{(2)} + \underbrace{\sum_{i=1}^m \mu_i^2 r_{M_i}}_{(3)}$$

For the given $S(n, k, m, l)$ and (σ_i^2, μ_i^2) , term (3) of (6.14) is constrained by the R_M requirement ($E_{\omega} u_i^2 = \mu_i^2 \forall i = 1, \dots, m$) to be a fixed value. Also, term (2) of (6.14) is constrained to be a fixed value by the Q_M^* requirement ($E_{\omega} y_i^2 \geq \sigma_i^2$). Given the above conditions on (2) and (3), the LQG controller will force (1) of (6.14) to be as small as possible. Hence, point c) of Theorem 4 is guaranteed.

The fact that term (1) of (6.14) is as small as possible for the given specifications also implies that if term (1) is not zero then a CVLQG solution does not exist. From the definition of Q_M^* , R_M and $\nu_{\text{spc}}^y(Q, R)$, $\nu_{\text{spc}}^y(Q_M^*, R_M) > k$ and [term (1) of (6.14)] > 0 are equivalent conditions. Therefore point b) of Theorem 4 is established. ###

The results of Theorem 4 directly establish weighting conditions which satisfy the input constrained requirement of (6.1). Part (a) of theorem 4 establishes (Q^*, R_M) as the weighting condition necessary to satisfy (6.1a), and if all outputs are above specification at (Q^*, R_M) , the condition for (6.1b). Parts (b) and (a), on the other hand, establish (Q_M^*, R_M) as a weighting condition which satisfies (6.1b).

Theorem 5 addresses the output constrained requirement (6.2).

Theorem 5: Output-constrained Conditions

For a system of type $S(n,k,m,l)$ under the regulation of the LQG controller defined by (3.2)-(3.6) (diagonal Q, R) and the assumption that the matrices R_M^*, Q_M exist, the following holds:

$$(a) \quad \nu_{\text{spc}}^u(R_M^*, Q_M) \leq \nu_{\text{spc}}^u(R, Q) \quad \forall R, Q \text{ subject to the constraint that } E_{\infty} y_i^2 = \sigma_i^2 \quad \forall i = 1, \dots, k$$

$$(b) \quad \nu_{\text{spc}}^u(R_M^*, Q_M) > k \Rightarrow (\mu^2, \sigma^2) \text{ cannot be satisfied by an LQG controller}$$

(c) Given condition (b) the LQG controller designed by (R_M^*, Q_M) minimizes the following

$$\sum_{i=1}^m E_{\infty} u_i^2 / \mu_i^2 \quad \forall i: E_{\infty} u_i^2 > \mu_i^2$$

Proof: From definition 4, the controller defined by (R_M^*, Q_M) minimizes the cost functional:

$$(6.15) \quad \nu = \underbrace{\sum_{i=1}^m E_{\infty} u_i^2 / \mu_i^2}_{(1)} + \underbrace{\sum_{i=1}^k \sigma_i^2 q_{M_i}}_{(2)}$$

The rest of the proof for part (a) is the dual of the proof of part (a) of Theorem 4 where input terms are substituted for output terms and (6.13) is replaced by (6.15).

Again from definition 4, the controller defined by (R_M^*, Q_M) minimizes the cost functional:

$$\begin{aligned}
 (6.16) \quad v = & \underbrace{\sum_{i=1}^m E_{\infty} u_i^2 / \mu_i^2}_{(1)} + \underbrace{\sum_{i=1}^m \mu_i^2 r_{M_i}^*}_{(2)} + \underbrace{\sum_{i=1}^k \sigma_i^2 q_{M_i}}_{(3)}
 \end{aligned}$$

The rest of the proof is again the dual of the proof of Theorem 4 where input terms are interchanged with output terms and (6.14) is replaced by (6.16). ###

6.3.3 Update Equations

While weighting conditions which satisfy (6.1) and (6.2), are specified in theorems 4 and 5, nothing is said about how to achieve the weighting conditions or if they exist. The iterative update equations used by the algorithm of section 6.4 to achieve the required weighting conditions is presented in this section. Section 6.3.4 discusses the existence question.

6.3.3.1 Input Update Equation. The input-constrained condition of (6.1) requires the following condition

$$(6.17a) \quad E_{\infty} u_i^2 = \mu_i^2 \quad \forall i = 1, \dots, m$$

or

$$(6.17b) \quad r_i^{-2} b_i^T K \hat{X} K b_i = \mu_i^2 \quad \forall i = 1, \dots, m$$

The requirement of (6.17) is a nonlinear programming problem within the larger CVLQG non-linear programming problem. The goal is to adjust r_i , $i = 1, \dots, m$ *without* using gradient techniques which require complex

calculations. Assume for a moment that (6.17) is satisfied. Then, the following algebraic manipulations are permissible:

$$(6.18) \quad r_i^2 = \frac{b_i^T \hat{K} \hat{X} K b_i}{\mu_i^2} \quad ; \quad i = 1, \dots, m$$

Multiplying and dividing the right-hand side of (6.18) by r_i^{-2} and substituting in $E_\infty u_i^2$ gives:

$$(6.19) \quad r_i^2 = \frac{E_\infty u_i^2}{\mu_i^2} r_i^2 \quad ; \quad i = 1, \dots, m$$

taking the positive square root of (6.19) leaves:

$$(6.20) \quad r_i = \left[\frac{E_\infty u_i^2}{\mu_i^2} \right]^{1/2} r_i \quad ; \quad i = 1, \dots, m$$

Equation 6.20 represents a an m-dimensional, coupled transcendental equation. A simple approach to numerically solving (6.20) is to adopt the following successive approximation equation:

$$(6.21) \quad r_i(j+1) = \left[\frac{E_\infty u_i^2(j)}{\mu_i^2} \right]^{1/2} r_i(j) \quad ; \quad i = 1, \dots, m$$

where $E_\infty u_i^2(j)$ implies $E_\infty u_i^2$ calculated at the j^{th} iteration. The beauty of (6.21) is that it will always correct $r_i(j)$ in the right direction. For instance, assume that $E_\infty u_i^2 > \mu_i^2$ for $r_i(j)$. To correct this situation in the LQG controller, the penalty weight $r_i(j)$ must be increased, and since $E_\infty u_i^2(j) > \mu_i^2$, this is exactly what (6.21)

does. When (6.21) was tested on the hoop-column and telescope models, the convergence to R_M was slow which is generally true of most successive approximation equations. However in each iteration of (6.21) an algebraic Riccati equation of order n must be solved and therefore, the slow convergence of (6.21) was deemed unacceptable.

Several options exist for increasing the step size of (6.21). The more sophisticated techniques involve iterations within each iteration to pin down the best step size to use and, of course, this is exactly what needs to be avoided. After, considerable testing the following automated non-linear step size adjustment was found to work 'well' on both the hoop column and telescope models:

$$(6.22) \quad r_i(j+1) = \left[\frac{E_{\infty} u_i^2(j)}{\mu_i^2} \right]^{PWR(j)} r_i(j) ; \quad i = 1, \dots, m$$

where the exponent $PWR(j)$ obeys the sequence:

$$(6.23) \quad PWR(1) = 1/2, \quad PWR(2) = 1, \dots, PWR(j) = j - 1 \quad \forall j \geq 1.$$

The problem with (6.22) is that it is now possible for the step size (i.e. change in $r(j)$) to be too big which would cause the algorithm to oscillate (i.e. not converge) or even go unstable. To counter this problem, the well known non-linear programming technique of descent functions was adopted.[60] The function chosen was:

$$(6.24) \quad Desctu(j+1) = \text{Max} \left[\left(\frac{E_{\infty} u_i^2(j+1)}{\mu_i^2} - 1 \right), 0 \right] \quad i = 1, \dots, m$$

Therefore, at each iteration $\text{Desctu}(j+1)$ is calculated. If $\text{Desctu}(j+1) > \text{Desctu}(j)$, then the current values of $\text{Desctu}(j+1)$, $E_{\omega}^2(j+1)$ are replaced with the old values $\text{Desctu}(j)$, $E_{\omega}^2(j)$ and the $\text{PWR}(j)$ sequence is reset to $\text{PWR}(1)$. For notational purposes, the iteration at which $\text{Desctu}(j+1) > \text{Desctu}(j)$ will be called a 'reset iteration.'

The two key questions for update equation (6.22) coupled with the descent function (6.24) are:

- (1) Is $\text{PWR}(1) = 1/2$ truly a conservative step size?
- (2) How often must data be lost to a 'reset iteration?'

Both questions actually apply to the sequence $\text{PWR}(j)$. Appendix C offers an argument for the conservative nature of $\text{PWR} = 1/2$ and for the hoop-column and telescope data shown in sections 6.5 and 6.6, less than 7% was lost because of reset iterations. There is, however no guarantee the chosen $\text{PWR}(j)$ sequence is the best, other possibilities are suggested in Chapter 8.

6.3.3.2 Output Update Equation. In contrast to the input-constrained requirement, the output constrained requirement of (6.2) requires the following condition:

$$(6.25a) \quad E_{\omega} y_i^2 = \sigma_i^2 \quad \forall i = 1, \dots, k$$

or using the identity of (5.7)

$$(6.25b) \quad v_i^y q_i^{-1} = \sigma_i^2 \quad \forall i = 1, \dots, k$$

Assuming that (6.25b) holds, the following algebraic manipulations are possible:

$$(6.26) \quad q_i = v_i^y / \sigma_i^2 \quad \forall i = 1, \dots, k$$

Again making use of (5.7):

$$(6.27) \quad q_i = \frac{E_{\omega} y_i^2}{\sigma_i^2} q_i \quad \forall i = 1, \dots, k$$

This equation (similar to (6.20)) represents a k-dimensional, coupled transcendental equation and as in the input case, a simple successive approximation technique was chosen to obtain a numerical solution:

$$(6.28) \quad q_i(j+1) = \frac{E_{\omega} y_i^2(j)}{\sigma_i^2} q_i(j) ; \quad i = 1, \dots, k$$

Like (6.21), (6.28) will always adjust $q(j)$ in the right direction; however, when tested it too demonstrated very slow convergence properties. Benefiting from the knowledge gained in the input case, the following update equation and descent function were adopted:

$$(6.29a) \quad q_i(j+1) = \left[\frac{E_{\omega} y_i^2(j)}{\sigma_i^2} \right]^{PWR_Y(j)} q_i(j) \quad i = 1, \dots, k$$

$$(6.29b)^{\dagger} \quad PWR_Y(1) = 1, \quad PWR_Y(2) = 1, \dots, PWR_Y(j) = j - 1, \quad j > 1$$

[†]The $PWR(j)$ sequence of (6.23) was also used successfully, however (6.29b) performed better for the chosen examples and is more in harmony with the development of (6.25)-(6.28).

$$(6.30) \quad \text{Descty}(j+1) = \text{Max} \left[\left(\frac{E_{\omega} y_i^2(j+1)}{\sigma_i^2} - 1 \right), 0 \right] ; i = 1, \dots, k$$

As in the input case, if $\text{Descty}(j+1) > \text{Descty}(j)$, then the current values of $\text{Descty}(j+1)$, $E_{\omega} y_i^2(j+1)$ are replaced with the old values $\text{Descty}(j)$, $E_{\omega} y_i^2(j)$ and the $\text{PWR}_Y(j)$ sequence is reset to $\text{PWR}_Y(1)$.

The same questions that exist for $\text{PWR}(j)$ also exist for $\text{PWR}_Y(j)$. Empirical results have pointed to the selection of $\text{PWR}_Y(j)$, but the suggested sequences in Chapter 8 offer other possibilities.

6.3.3.3. Simultaneous Input and Output Updates. When conditions require a search for the matrices Q_M^* or R_M^* , simultaneous updates of Q and R are required. The transcendental nature of the problem remains the same. Therefore, (6.22) is used for the input weights and (6.29a) for the output weights. Both descent functions are still calculated but combined as follows:

$$(6.31) \quad \text{Desctyu}(j+1) \triangleq \text{Max}[\text{Desctu}(j), \text{Descty}(j), 0]$$

If a 'reset iteration' occurs *all* input and output values are restored as well as $\text{Desctyu}(j)$.

As a final summary, the update equations of (6.22) and (6.29a) are modified successive approximations of the transcendental equations (6.20) and (6.27). They will *always* step in the right direction and the descent function is used to insure they never step too far. The $\text{PWR}(j)$, $\text{PWR}_Y(j)$ exponent sequences are incorporated in an effort to prevent an excessive number of iterations for convergence.

6.3.4 Existence of R_M , Q_M , Q_M^* , R_M^*

The final question to be answered concerning the theory behind the CVLQG algorithm presented in 6.4 (entitled LQGWTS) is whether the weighting matrices R_M , Q_M , Q_M^* , R_M^* exist. The question can be simplified by realizing that, by definition (i.e. Definition 4), if R_M exists R_M^* exists and if Q_M exists Q_M^* exists. Therefore, attention will be focused on the existence of R_M and Q_M .

6.3.4.1 Existence of R_M . An R_M *cannot* exist if the following condition holds for some LQG control u_i :

$$(6.32a) \quad E_{\infty} u_i^2 = r_i^{-2} b_i^T \hat{K} \hat{X} K b_i = 0 \quad \forall \quad Q > 0, \quad R > 0,$$

or since $0 < r_i < \infty$,

$$(6.32b) \quad b_i^T \hat{K} \hat{X} K b_i = 0 \quad \forall \quad Q > 0, \quad R > 0.$$

Necessary and sufficient conditions for (6.32b) to hold involve the observability of (A,C) and the controllability of $(A+BG,F)$ and are provided in Appendix D. Also included in Appendix D is the method LQGWTS uses to determine if u_i satisfies the necessary and sufficient conditions for (6.32b). If (6.32b) does not hold for all i then an R_M exists for arbitrary non-zero specifications (u^2). More specifically, if (6.32b) does not hold, it is possible to force all controls to arbitrary values $E_{\infty} u_i^2$ by weighting certain controls more than others. Furthermore, since the u_i 's that satisfy (4.9b) are of *no use* to an LQG controller, condition (6.32b) can *always* be avoided by deleting from the system controls which satisfy (6.32b), and LQGWTS does this.

Therefore LQGWS seeks, a possibly reduced dimension, R_M (i.e. R_{M_R}) which is known to exist.

6.3.4.2 Existence of Q_M . The existence of Q_M depends upon the ability of the system to drive all outputs to arbitrary levels. For controllable systems, this implies the outputs ($y = Cx$) must be linearly independent. For un-controllable systems, this implies that the rank of the output controllability matrix must equal k (the number of outputs). [63]. If, for a system of type $S(n,k,m,\ell)$, Q_M fails to exist for either of the above reasons, the end result is that only some of the outputs will be driven to their specification. Label this set of outputs Y_{spc} and its reduced dimension weighting matrix Q_{MR} . The remaining outputs will be in the set \bar{Y}_{spc} . The weights on these outputs will have one of the two following properties as a result of using the update equation (6.29a):

$$(6.33) \quad \begin{cases} q_i \rightarrow 0 \\ q_i \rightarrow \infty^\dagger \end{cases}$$

The situation in (6.33) means that the outputs in \bar{Y}_{spc} are *not* effecting the LQG regulator design and *may* be deleted from the system leaving only the set Y_{spc} for which a Q_{MR} exists. LQGWS tests for outputs which are not effecting the design by performing the checks

[†]The condition $q_i \rightarrow \infty$ can result only when a stabilizable, detectable system is *not* output controllable and one of the uncontrollable outputs stabilizes at a variance level above its specification.

$$(6.34a) \quad q_i \stackrel{?}{<} \epsilon / \sigma_i$$

$$; \quad 0 < \epsilon \ll 1$$

$$(6.34b) \quad q_i \stackrel{?}{>} \frac{1}{\epsilon \sigma_i}$$

If (6.34a or b) is satisfied LQGWS automatically zeros q_i (i.e. effectively deletes the i^{th} output from the system). Therefore LQGWS will look for a Q_{MR} which exists.

6.3.5 Summary

The purposes of this section have been first, to establish the weighting conditions which satisfy the requirements of (6.1) and (6.2) (i.e. Theorems 4 and 5). These conditions are using the notation of Section 6.3.4 R_{MR} , Q_{MR} , R_{MR}^* , and Q_{MR}^* . Second, to develop the update equations used by LQGWS and verify that they converge to R_{MR} , Q_{MR} , R_{MR}^* or Q_{MR}^* , and thirdly, as just presented, verify the existence of these matrices. With this theory in place, a presentation of the algorithm is in order.

6.4 The Algorithm LQGWS

This section outlines the proposed algorithm LQGWS which is designed to find a solution to the CVLQG problem defined at the beginning of this chapter. As already noted, LQGWS uses the update equations (6.22) and/or (6.29a) to achieve either the input constrained requirements of (6.1) or the output-constrained requirements of (6.2). The general flow of the algorithm for achieving these requirements can be summarized by the following two line graphs which delineate the weight adjustment procedure:

Input-Constrained Algorithm

$$(6.35a) \quad (Q^*, R^*) \longrightarrow (Q^*, R_M) \longrightarrow (Q_M^*, R_M^*)$$

$$\text{Begin} \longrightarrow \text{Solution?} \text{-----} \text{End}$$

Output-Constrained Algorithm

$$(6.35b) \quad (R^*, Q^*) \longrightarrow (R^*, Q_M) \longrightarrow (R_M^*, Q_M^*)$$

$$\text{Begin} \longrightarrow \text{Solution?} \text{-----} \text{End}$$

With the general flow of (6.35) in mind, the concise steps of the algorithm are presented.

Algorithm LOGWTS:

- ①. Read the following data: $\{A, B, C, D, M, W, V^{-1}\}$, $\{\sigma_i^2, i = 1, \dots, k\}$ $\{\mu_i^2, i = 1, \dots, m\}$, select input-constrained or output constrained option, and specify a zero-threshold parameter ϵ .
- ②. Compute P by solving (3.2d).
- ③. Compute $c_i^T P c_i$, $i = 1, \dots, k$. If $c_i^T P c_i \geq \sigma_i^2$ for any $i = 1, \dots, k$, it is impossible to achieve the output specifications since $c_i^T P c_i$ is the lower bound of $E_{\omega_i}^2$. Therefore select only the-input constrained option.
- ④. Set the following initial guess for Q and R :

$$q_i = 1/\sigma_i^2; \quad i = 1, \dots, k \quad (\text{i.e. } q_i^*)$$

$$r_i = 1/\mu_i^2; \quad i = 1, \dots, m \quad (\text{i.e. } r_i^*)$$

- ⑤. Compute K by solving (3.2c), and \hat{X} by solving (3.6). Also as described in Appendix D, perform first iteration check for $b_i^T \hat{K} \hat{X} K b_i = 0$.
- ⑥. Compute $E_{\omega y_i}^2$, $i = 1, \dots, k$ using (3.16). Compute $E_{\omega u_i}^2$, $i = 1, \dots, m$ using (3.19). Perform second check described in Appendix D for $b_i^T \hat{K} \hat{X} K b_i = 0$. Also perform the following:
- (a) Input-Constrained Search
 If $E_{\omega u_i}^2 = u_i^2 \quad \forall i = 1, \dots, m$, then store for later use the indices j for which $E_{\omega y_j}^2 < \sigma_j^2$. Call this vector of indices UPDATE.
- (b) Output-Constrained Search
 If $E_{\omega y_i}^2 = \sigma_i^2 \quad \forall i: q_i \neq 0$, then store for later use the indices j for which $E_{\omega u_j}^2 < \mu_j^2$. Call this vector of indices UPDATE.
- ⑦. Calculate Descent Function (DESCTN)
- (a) Input-constrained search: use (6.24)
- (b) Output constrained search: use (6.30)
- (c) If UPDATE $\neq 0$ use (6.31).
- ⑧[†]. Check DESCNTN:
- If $\text{DESCNTN}(j+1) > \text{DESCNTN}(j)$ set $\text{PWR}(j+1) = 1/2$, $\text{PWR}_y(j+1) = 1$,
 $E_{\omega y_i}(j+1) = E_{\omega y_i}(j)$, $E_{\omega u_i}(j+1) = E_{\omega u_i}(j)$ $\text{DESCNTN}(j+1) =$
 $\text{DESCNTN}(j)$ and go to ⑩. (i.e. 'reset iteration').

[†]For numerical considerations, the following additional reset condition is defined. If $\text{DESCNTN}(j+1) > \text{DESCNTN}(j)$ and $\text{DESCNTN}(j) = 0$, reset *only* PWR and PWR_y. This condition is termed 'step size' reset to distinguish it from a 'reset iteration'!

⑨ . Solution Checks:

Input-Constrained Option:

C.1 If $E_{\infty} u_i^2 = \mu_i^2 \forall i = 1, \dots, m$, and $\{q_i = 0 \forall i: E_{\infty} y_i^2 < \sigma_i^2\}$
then the CVLQG input solution has been found. Stop.

Output-Constrained Option:

C.2 If $\{E_{\infty} y_i^2 = \sigma_i^2 \forall i: q_i \neq 0\}$, and $E_{\infty} u_i^2 \geq \mu_i^2 \forall i = 1, \dots, m$, then the CVLQG output solution has been found.
Stop.

⑩ . Update Equations: Define $PWR(j+1)$ as in (6.23) and $PWR_Y(j+1)$
as in (6.29b)

Input Constrained Option:

$$r_i(j+1) = \left[\frac{E_{\infty} u_i^2(j)}{\mu_i^2} \right]^{PWR(j+1)} r_i(j); \quad i = 1, \dots, m$$

$$q_i(j+1) = \left[\frac{E_{\infty} y_i^2(j)}{\sigma_i^2} \right]^{PWR_Y(j+1)} q_i(j); \quad \forall i \text{ in UPDATE}$$

$$q_i(j+1) = q_i(j) \text{ otherwise}$$

If $q_i < \epsilon/\sigma_i^2$ or $q_i > \frac{1}{\epsilon\sigma_i^2}$ then set $q_i(j+1) = 0$.

Return to ⑤.

Output-Constrained Option:

$$(3.4a) \quad q_i(j+1) = \left[\frac{E_{\infty} y_i^2(j)}{\sigma_i^2} \right]^{PWR_Y(j+1)} q_i(j); \quad i = 1, \dots, k$$

If $q_i < \epsilon/\sigma_i^2$ or $q_i > \frac{1}{\epsilon\sigma^2}$ then set $q_i(j+1) = 0$

$$(3.4b) \quad r_i(j+1) = \left[\frac{E_{\infty} u_i(j)}{\mu_i^2} \right]^{PWR(j+1)} r_i(j) ; \quad \forall i \text{ in UPDATE}$$

$$r_i(j+1) = r_i(j) \text{ otherwise}$$

Return to ⑤.

The algorithm LQGWTS has been written as a fortran IV subroutine which uses the linear control package LSLIB developed for the Purdue University CDC 6600 by C.S. Gregory in 1979. The specific details of the program, along with a program listing are provided in Appendix E. The subroutine has been used on both the hoop column antenna and telescope models of Chapter 4. The results are presented in the next two sections.

6.5 Hoop Column Antenna Example

The algorithm LQGWTS was run on the hoop column antenna model $S_{\text{Hoop}}(26,24,12,39)$ for the outputs, sensors, and actuators described in tables 4.1-4.3 and for the (σ^2, μ^2) specifications of (4.16). Both the input-constrained and output-constrained options were run, and the results indicated that an LQG controller (diagonal Q,R) did not exist to meet (σ^2, μ^2) of (4.16). Therefore, the 'minimum achievable' specifications of (6.1b) and (6.2b) were sought.

6.5.1 Input-Constrained Solution

Table 6.1 displays the results for the input constrained solution. The expression in parentheses by each output and actuator number is a

Table 6.1: Hoop Column Input-Constrained Solution

Output #	$\sqrt{E_{\omega} y_i^2}$ (minimum achievable)	Actuator #	$\sqrt{E_{\omega} u_i^2}$ (specification)
1 (AX2)	.171 sec ($q_1 = 0$)	1 (TX2)	10.000 dn-cm
2 (AY2)	.174 sec ($q_2 = 0$)	2 (TY2)	"
*3 (AZ2)	701.807 sec	3 (TZ2)	"
4 (AX10-AX2)	.008 sec ($q_4 = 0$)	4 (TX6)	"
5 (AY10-AY2)	.008 sec ($q_5 = 0$)	5 (TY6)	"
*6 (AZ10)	727.366 sec	6 (TZ6)	"
7 (X6-X2)	.122 mm ($q_7 = 0$)	7 (TX9)	"
8 (Y6-Y2)	.120 mm ($q_8 = 0$)	8 (TY9)	"
*9 (X9-X2)	.799 mm	9 (TZ9)	"
*10 (Y9-Y2)	.784 mm	10 (TX10)	"
*11 (X10-X2)	1.859 mm	11 (TY10)	"
*12 (Y10-Y2)	1.824 mm	12 (TZ10)	
*13 (X101-X10)	3.003 mm		
*14 (Y101-Y10)	7.595 mm		
15 (Z101-Z10)	.091 mm ($q_{15} = 0$)		
*16 (X107-X10)	7.219 mm		
*17 (Y107-Y10)	3.381 mm		
18 (Z107-Z10)	.090 mm ($q_{18} = 0$)		
*19 (X113-X10)	2.054 mm		
*20 (Y113-Y10)	3.474 mm		
21 (Z113-Z10)	.016 mm ($q_{21} = 0$)		
*22 (X119-X10)	3.728 mm		
*23 (Y119-Y10)	1.799 mm		
24 (Z119-Z10)	.010 mm ($q_{24} = 0$)		

* → specification violation ($E_{\omega} y_i^2 > \sigma_i^2$)

label for the type, direction, and location of the output or actuator. For instance, AX2 stands for an angle output in the X direction at node 2, AX10-AX2 represents a relative angle output in the X direction between nodes 10 and 2, Y6-Y2 means a relative linear displacement output in the Y direction between nodes 6 and 2 and TX2 stands for a torquer acting in the X direction at node 2. The output weight in parentheses next to a specification represents the weight that the algorithm assigned to that output on the final iteration. For the given (σ^2, μ^2) , these outputs form the set \bar{Y}_{spc} and have no effect on the LQG regulator. As expected, the algorithm forced all actuators to operate at their specification (i.e. 10 dn-cm). The data shown in Table 6.1 is the result of 16 iterations of the input-constrained search and two 'step size' resets were required. The plots of Figures 6.4-6.6 show the *normalized* values of each input (i.e. $E_{\omega} u_i^2 / \mu_i^2$ versus the iteration number). The value 1.0 implies the component is in specification, and the value of the last data point is printed on each plot.

A striking feature of these plots is that the algorithm has essentially converged after 5-8 iterations yet continues for another 8-11 iterations. This results because in 8 iterations the algorithm converged to (Q^*, R_M) . It then took 8 more iterations to identify the outputs in \bar{Y}_C (i.e. $q_i \rightarrow 0$, $(Q^*, R_M) \rightarrow (Q_{MR}^*, R_M)$). This indicates that, for this example, the ϵ of .001 selected for the algorithm and used in setting the $q_i \rightarrow 0$ threshold was smaller than necessary. Since (Q^*, R_M) did converge to (Q_{MR}^*, R_M) the output specifications of Table 5.4 are the minimum specifications promised by Theorem 4 and required by (6.1b).

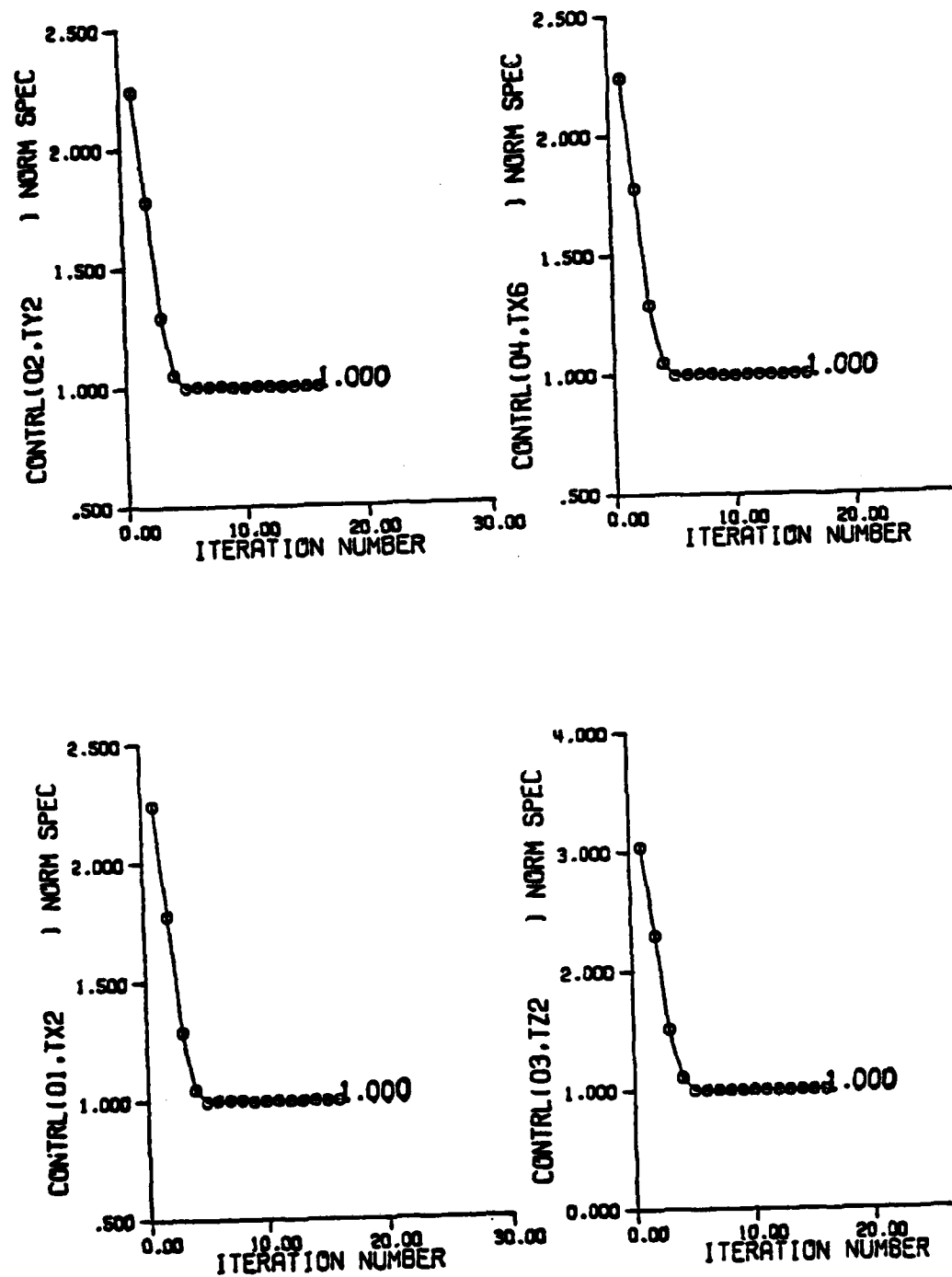


Figure 6.4: Hoop Column Input-Constrained Solution (actuators 1-4)

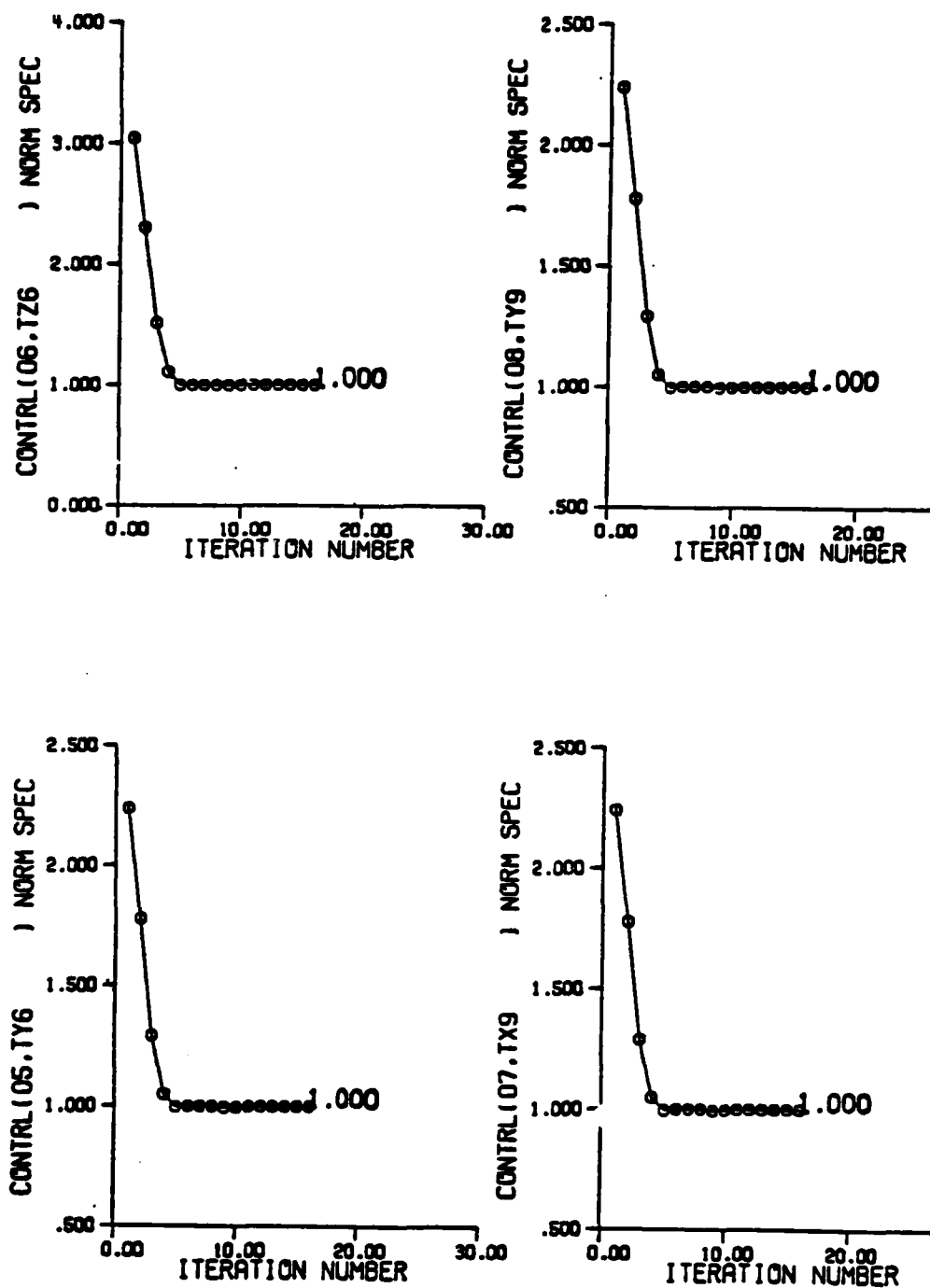


Figure 6.5: Hoop Column Input-Constrained Solution (actuators 5-8)

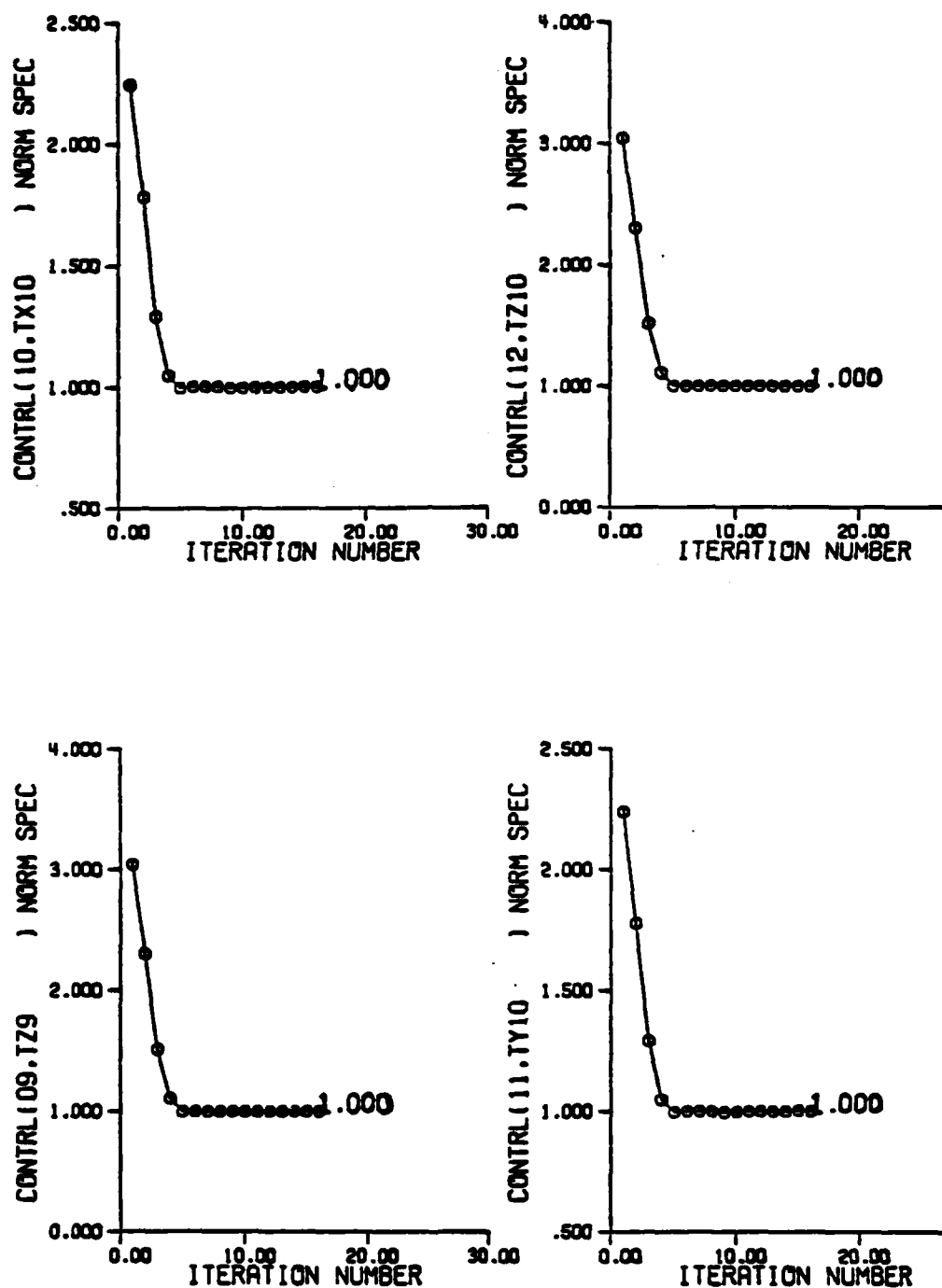


Figure 6.6: Hoop Column Input-Constrained Solution (actuators 9-12)

Figure 6.7 is a representative example of the behavior of 4 out of the 24 outputs during this input-constrained search. The plots in this figure also show *normalized* values versus the iteration number.

6.5.2 Output-constrained Solution

The output constrained search produced an LQG controller which satisfied the requirements of (6.2b). The results are shown in Table 6.2. The numbers in parentheses in Table 6.2, as in Table 6.1, represent the weight that the algorithm assigned to that particular output on the last iteration. As expected, the output-constrained search drove the outputs in \mathcal{Y}_C to their specification. It also assigned zero weights to all outputs in $\bar{\mathcal{Y}}_C$. Since the hoop-column model is controllable, these results indicate that the rank of the 24x26 C matrix is only 7 and for the particular (σ^2, μ^2) outputs 11, 12, 14, 16, 17, 20, and 22 form the independent set. The input-specifications shown in Table 6.2 represent the minimum torque specifications required by (6.2b) and promised in Theorem 5. The data is the result of 18 iterations of the output-constrained search *plus* two 'reset iterations' (i.e. PWR reset to .5 and previous iteration data restored). Figures 6.8-6.13 display the *normalized* values of each output component (1.0 implies the component is at specification) versus the iteration number, and reset iterations are not shown. As in Figures (6.4-6.7), the last data point is printed on each plot. The apparent 'extra' iterations of the output-constrained search, as in the input-constrained search, are necessary to identify the output in \mathcal{Y}_C (i.e. $q_i \rightarrow 0 \quad \forall i: y_i \in \mathcal{Y}_C$) for the specified ϵ .

Figure 6.14 is a representative example of the behavior of 4 out of

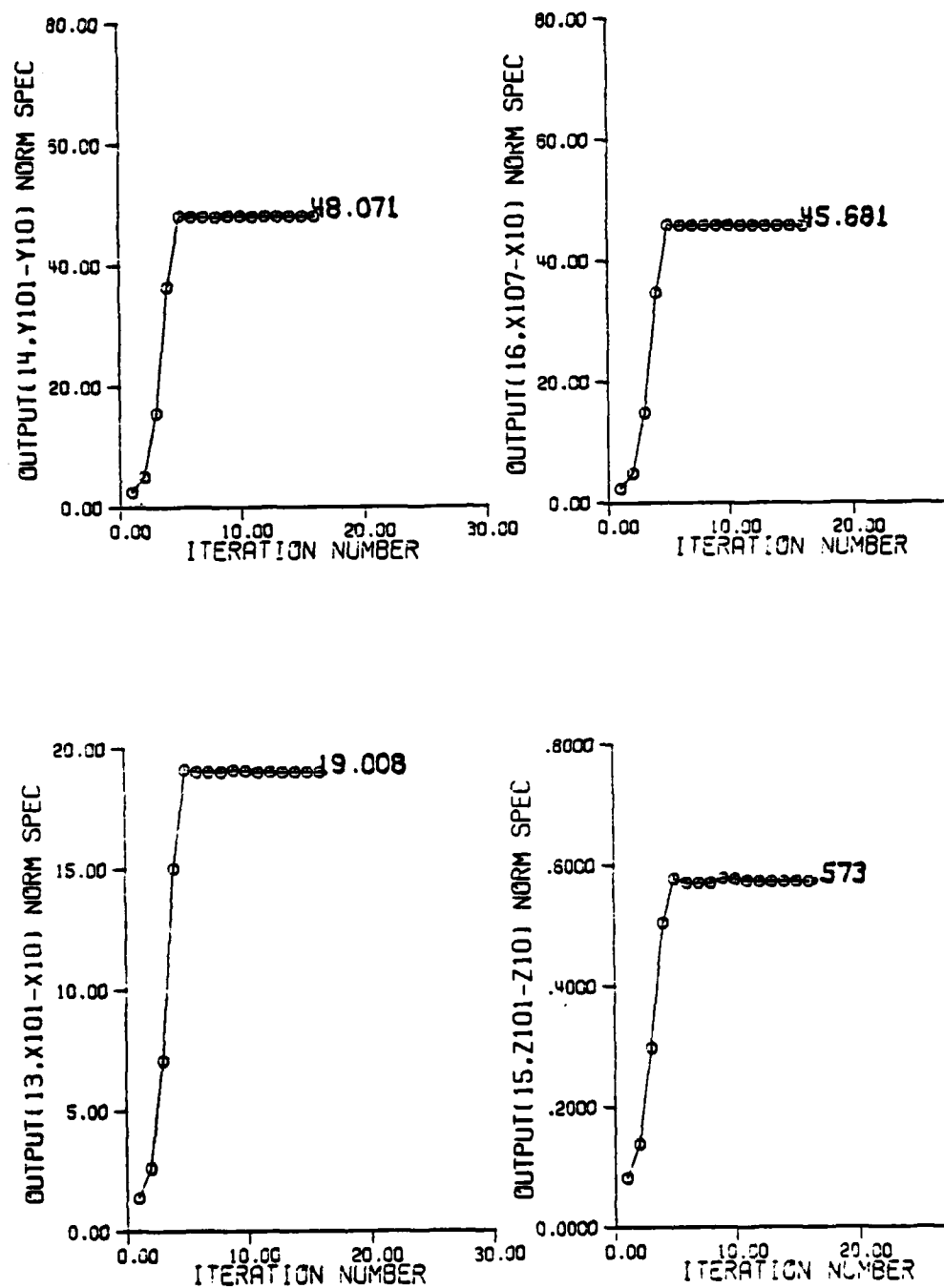


Figure 6.7: Hoop Column Input-Constrained Solution (outputs 13-16)

Table 6.2: Hoop Column Output-Constrained Solution Results

Output#	$\sqrt{E_{\infty} y_i^2}$		Actuator #	$\sqrt{E_{\infty} u_i^2}$	(minimum achievable)
1(AX2)	.015 sec	($q_1 = 0$)	*1(TX2)	24.252	dn-cm
2(AY2)	.015 sec	($q_2 = 0$)	*2(TY2)	24.288	dn-cm
3(AZ2)	11.579 sec	($q_3 = 0$)	*3(TZ2)	40.280	"
4(AX10-AX2)	.001 sec	($q_4 = 0$)	*4(TX6)	24.253	"
5(AY10-AY2)	.001 sec	($q_5 = 0$)	*5(TY6)	24.282	"
6(AZ10)	12.001 sec	($q_6 = 0$)	*6(TZ6)	40.869	"
7(X6-X2)	.010 mm	($q_7 = 0$)	*7(TX9)	29.466	"
8(Y6-Y2)	.010 mm	($q_8 = 0$)	*8(TY9)	29.496	"
9(X9-X2)	.068 mm	($q_9 = 0$)	*9(TZ9)	41.963	"
10(Y9-Y2)	.068 mm	($q_{10} = 0$)	*10(TX10)	36.026	"
11(X10-X2)	.158 mm		*11(TY10)	36.056	"
12(Y10-Y2)	.158 mm		*12(TZ10)	41.747	"
13(X101-X10)	.104 mm	($q_{13} = 0$)			
14(Y101-Y10)	.158 mm				
15(Z101-Z10)	.007 mm	($q_{15} = 0$)			
16(X107-X10)	.158 mm				
17(Y107-Y10)	.156 mm				
18(Z107-Z10)	.008 mm	($q_{18} = 0$)			
19(X113-X10)	.122 mm	($q_{19} = 0$)			
20(Y113-Y10)	.159 mm				
21(Z113-Z10)	.001 mm	($q_{21} = 0$)			
22(X119-X10)	.158 mm				
23(Y119-Y10)	.091 mm	($q_{23} = 0$)			
24(Z119-Z10)	.001 mm	($q_{24} = 0$)			

* \Rightarrow specification violation ($E_{\infty} u_i^2 > \mu_i^2$)

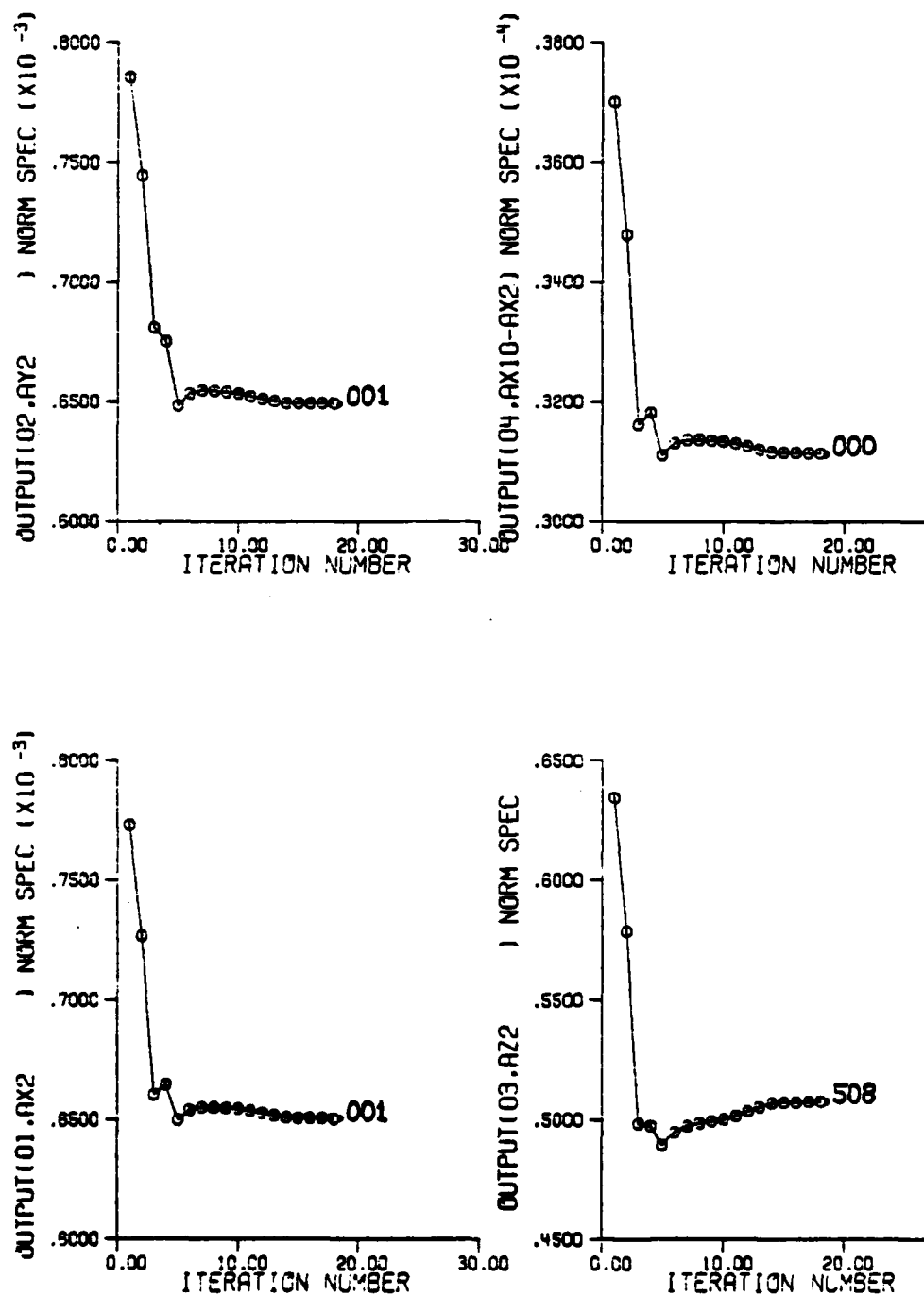


Figure 6.8: Hoop Column Output-Constrained Solution (outputs 1-4)

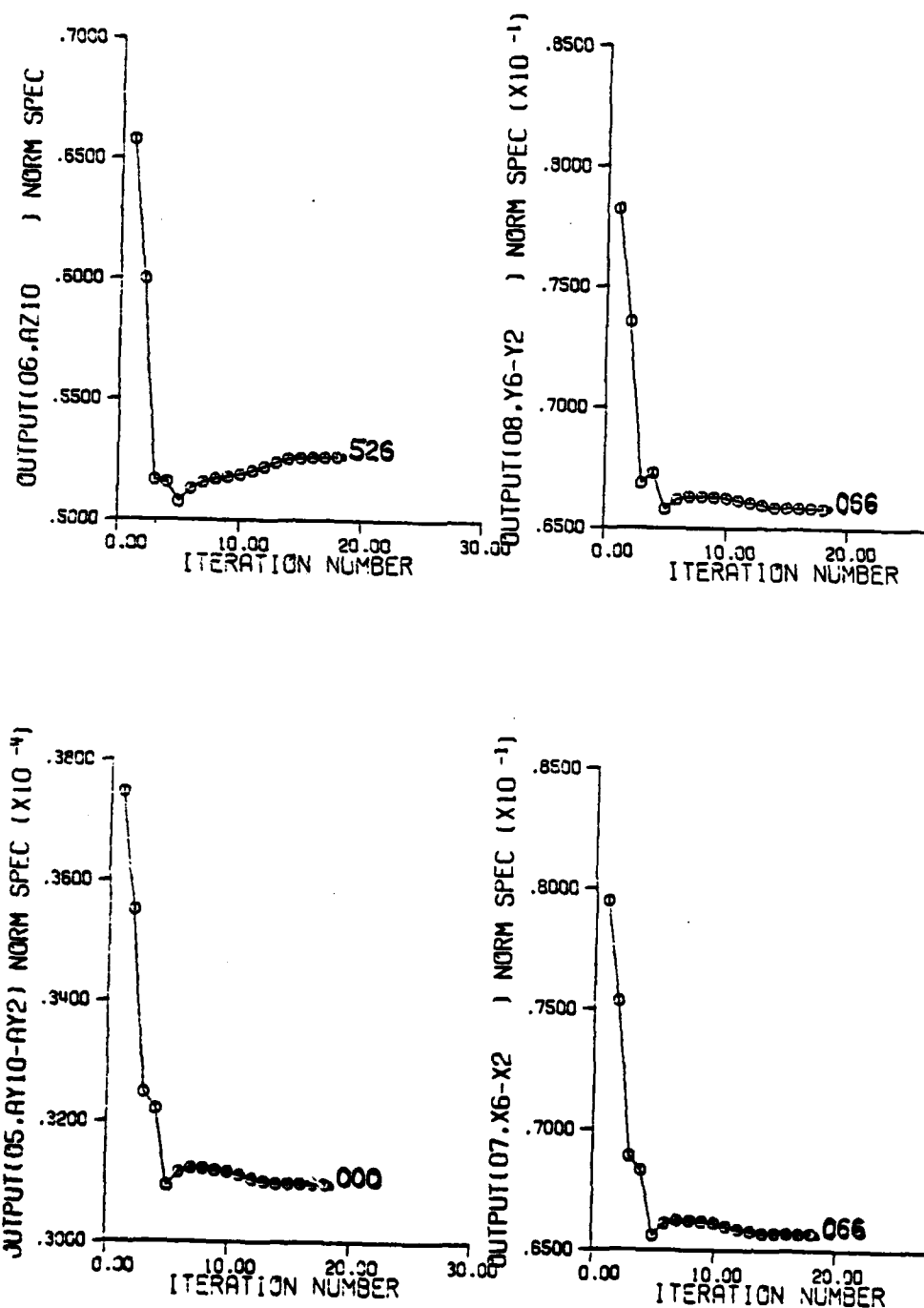


Figure 6.9: Hoop Column Antenna Output-Constrained Results (outputs 5-8)

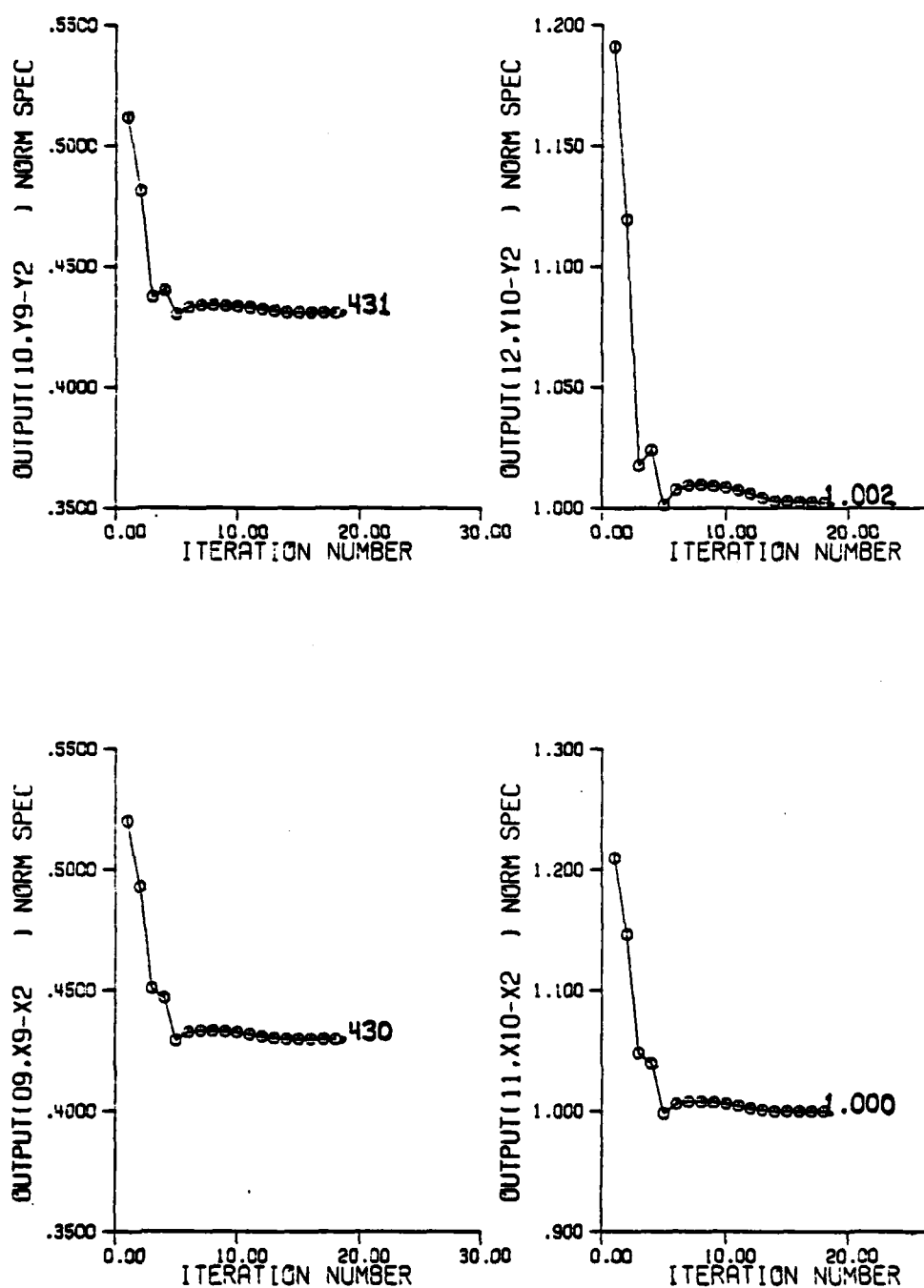


Figure 6.10: Hoop Column Output-Constrained Solution (outputs 9-12)

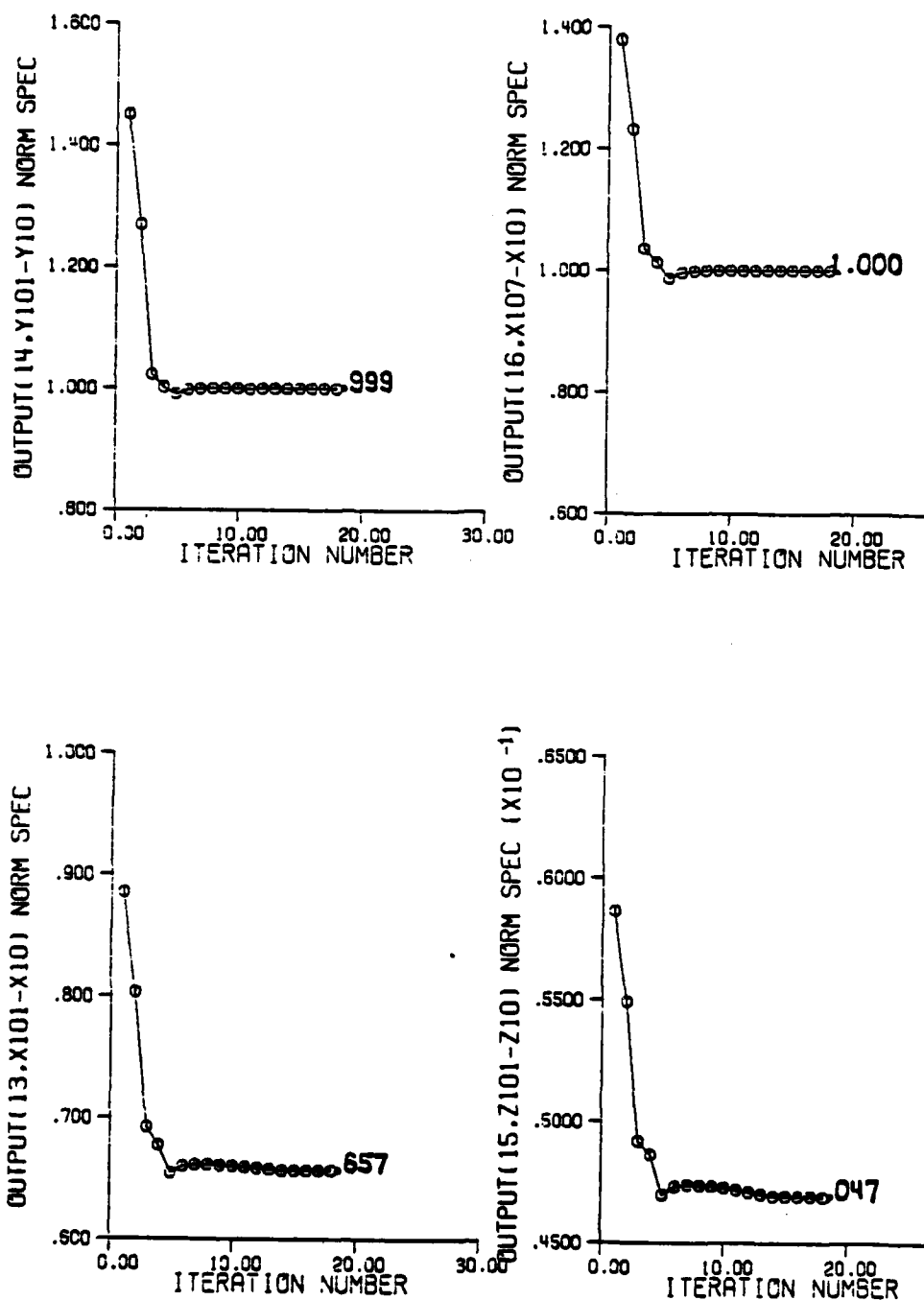


Figure 6.11: Hoop Column Output-Constrained Results (outputs 13-16)

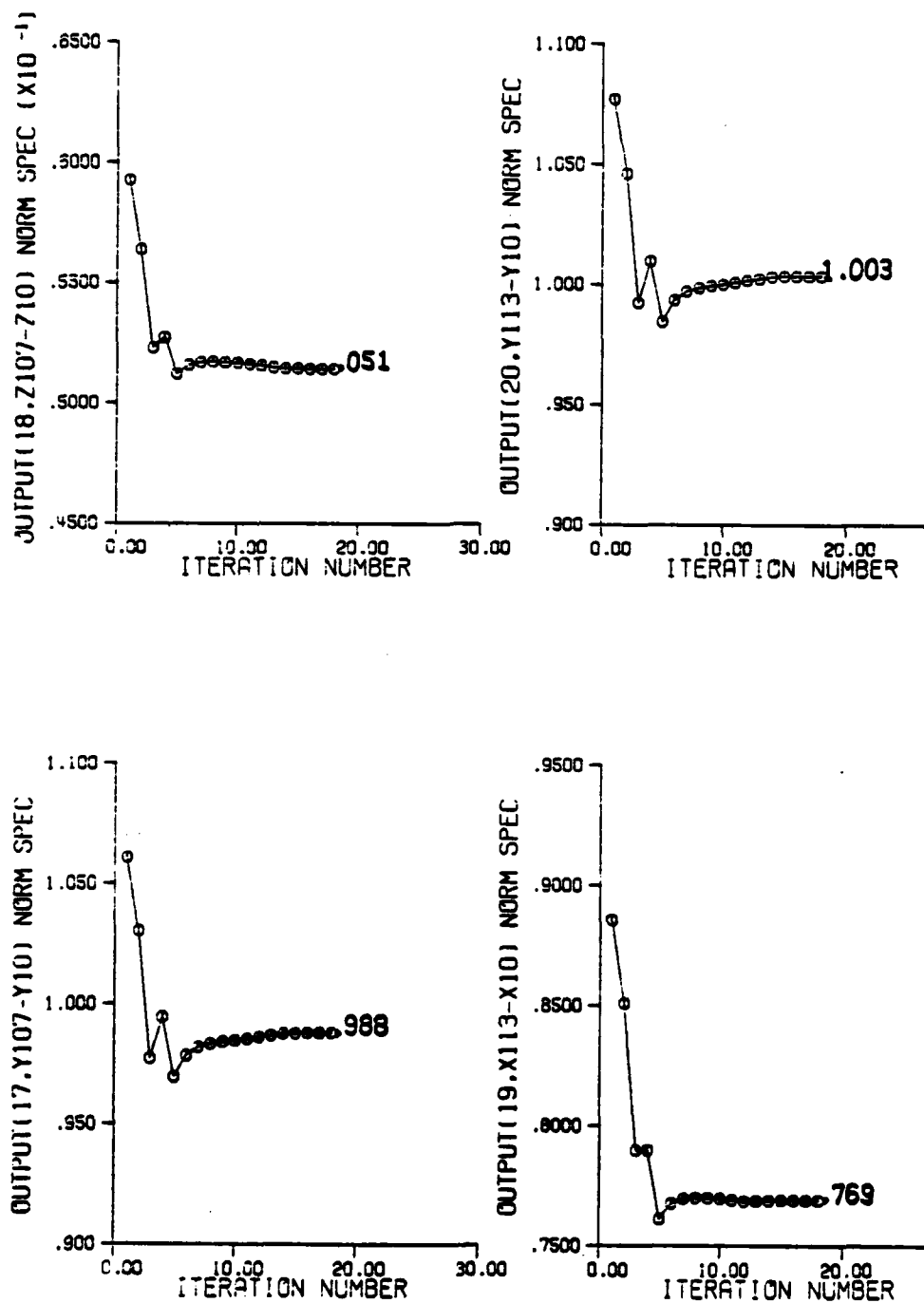


Figure 6.12: Hoop Column Output-Constrained Solution (outputs 17-20)

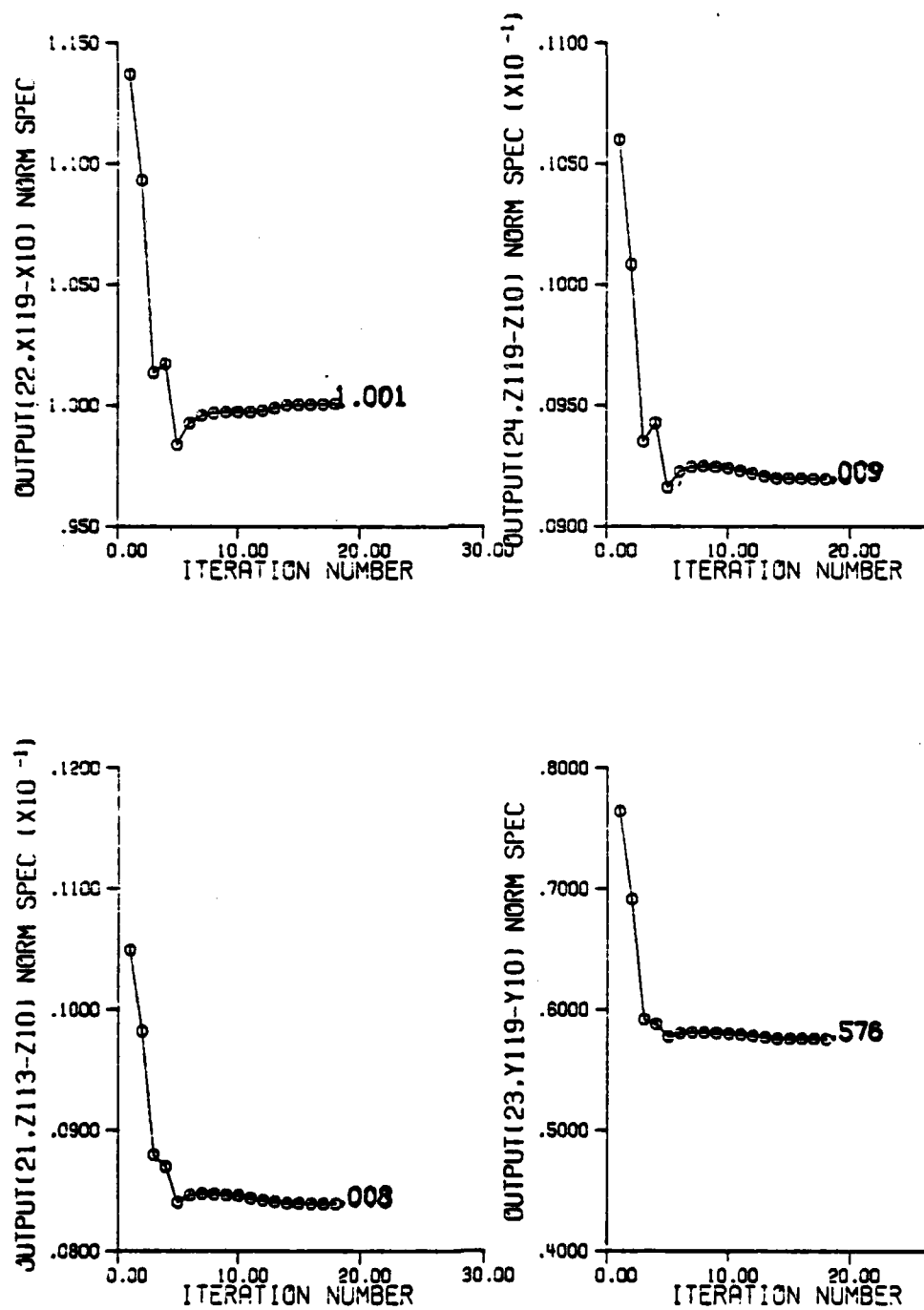


Figure 6.13: Hoop Column Output-Constrained Solution
(outputs 21-24)

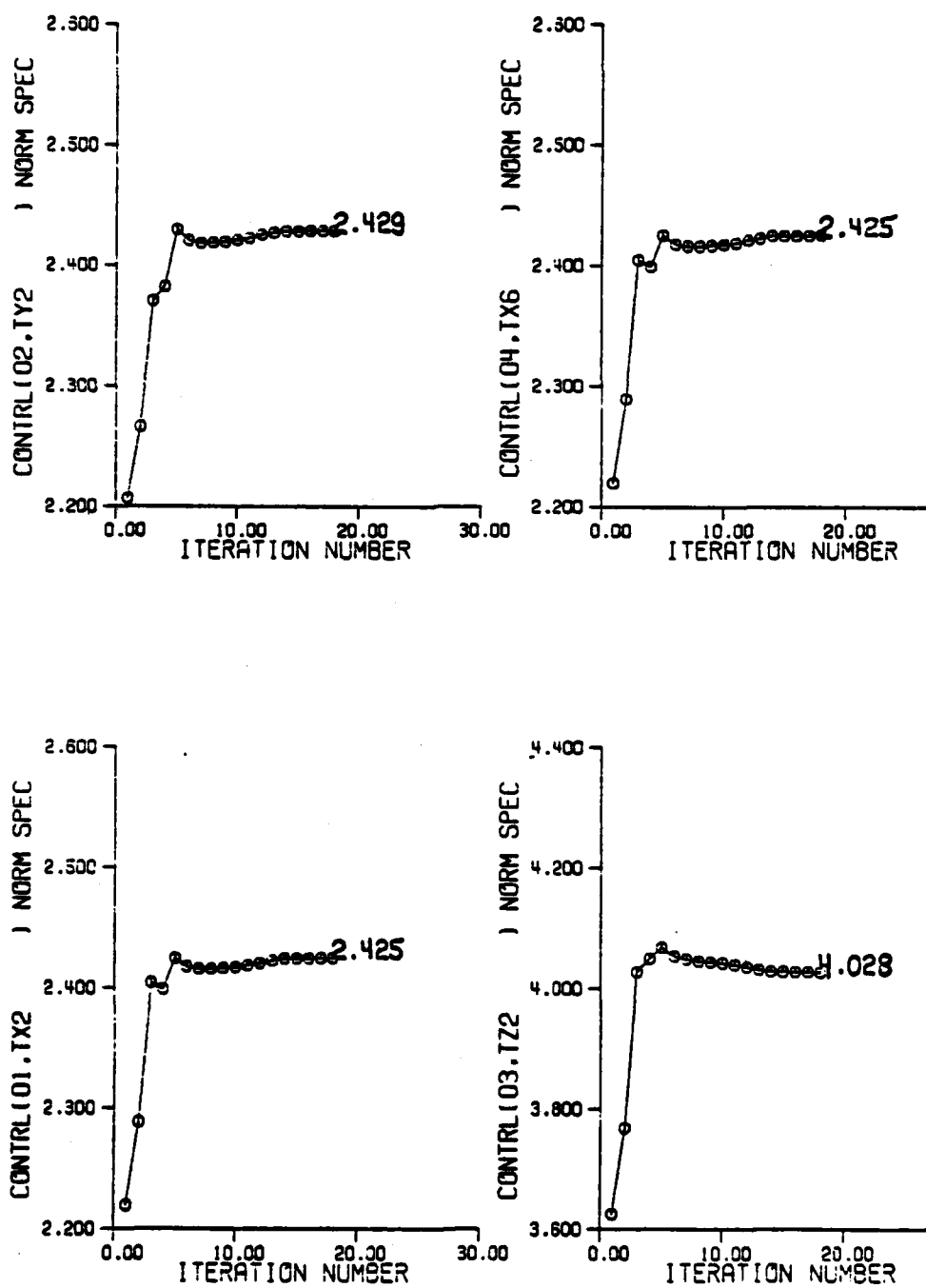


Figure 6.14: Hoop Column Output-Constrained Solution (actuators 1-4)

the 12 controls during the output-constrained search. The plots of this figure also display *normalized* values versus the iteration number.

6.5.3 Summary of Hoop Column Results

Physical insights into the control of the hoop-column antenna will be reserved for Chapter 7. Of interest in Section 6.5 has been the performance of LQGWS for the multi-input/output hoop model S_{Hoop} (26, 24, 12, 39). As evidenced by the data, the algorithm converges to the input-constrained or output-constrained solution. Considering the fact that for the input-constrained case, the algorithm is solving 22 simultaneous transcendental equations (12 input, 10 output), and for the output-constrained, case 17 simultaneous transcendental equations (17 output), the convergence rate is impressive. For a larger ϵ value, the rate would have been even faster. It should also be noted that only 2 out of the 36 iterations were lost due to iteration resets. Therefore, for the hoop column model, the modified successive approximation/descent function approach of LQGWS is practical and succeeds in tailoring the LQG cost functional to the input/output requirements of the SASLQG problem! The next section shows further evidence of the validity of LQGWS.

6.6 Solar Optical Telescope Example

The algorithm LQGWS was also run on the solar optical telescope model S_{tele} (24, 3, 21, 45) for the output, sensor, and actuator configuration described in Section 4.4 and the (σ^2, μ^2) specification of Table 4.14. As in the hoop case, both the input constrained and output constrained options were run, and the results indicated that an LQG controller (diagonal Q,R) did not exist to meet the (σ^2, μ^2)

of Table 4.14. Therefore, the 'minimum achievable' specifications of (6.1b) or (6.2b) were again sought.

6.6.1 Input-Constrained Solution

Table 6.3 displays the results for the input constrained solution, with appropriate labels by each output and actuator number. The algorithm placed a zero weight on the defocus output, and for the specifications of Table 4.14 the set \bar{Y}_c is the defocus output. As expected, the input-constrained solution required all actuators to operate at their specification, and the output values of Table 6.4 for the given actuator constraints represent the minimum achievable output specifications required by (6.1b) and promised by Theorem 4. The data of Table 6.3 is the result of 14 iterations of LQGWS plus two 'reset iterations.' The plots of figures 6.15-6.20 show the *normalized* value of each input and output component versus the iteration number (reset iterations are not shown), and the value of the last data point is printed on each plot. From looking at the plots, it appears that the algorithm converged by approximately iteration 9. As was the case in the hoop-example, the 'extra' iterations are required to identify, based on the specified ϵ threshold of .001, the outputs in \bar{Y}_c , and thus guarantee the minimum specifications promised by Theorem 4. (i.e. $(Q^*, R_M) \rightarrow (Q_{MR}^*, R_M)$).

6.6.2 Output-Constrained Solution

The output-constrained version of LQGWS produced an LQG regulator for the telescope which generated the data of Table 6.4.

Table 6.3: Telescope Input-Constrained Solution

output #	$\sqrt{E_{\omega y_i}^2}$ (minimum achievable)	actuator #	$\sqrt{E_{\omega u_i}^2}$ (specification)
*1(LOS X)	8.809 deg	1(FY1)	.01N
*2(LOS Y)	8.465 deg	2(FZ1)	"
3(DEFOCUS)	.003 mm ($q_3 = 0$)	3(FZ2)	"
		4(FX3)	"
		5(FY3)	"
		6(FZ3)	"
		7(FZ4)	"
		8(FX5)	"
		9(FY5)	"
		10(FZ5)	"
		11(FZ6)	"
		12(FY7)	"
		13(FZ7)	"
		14(FZ8)	"
		15(FZ9)	"
		16(FZ10)	"
		17(FX11)	"
		18(FY11)	"
		19(FZ11)	"
		20(FY12)	"
		21(FZ12)	"

* \Rightarrow specification violation ($E_{\omega y_i}^2 > \sigma_i^2$)

Table 6.4: Telescope Output Constrained Solution

output #	$\sqrt{E_{\omega y_i}^2}$	actuator #	$\sqrt{E_{\omega u_i}^2}$ (minimum achievable)
1(LOS X)	65.227 $\widehat{\text{sec}}$	*1(FY1)	.021N
2(LOS Y)	65.227 $\widehat{\text{sec}}$	*2(FZ1)	.030N
3(DEFocus)	.003 mm ($q_3 = 0$)	*3(FZ2)	.037N
		*4(FX3)	.041N
		*5(FY3)	.021N
		*6(FZ3)	.030N
		*7(FZ4)	.037N
		*8(FX5)	.208N
		*9(FY5)	.135N
		*10(FZ5)	.021N
		*11(FZ6)	.025N
		*12(FY7)	.135N
		*13(FZ7)	.021N
		*14(FZ8)	.025N
		*15(FZ9)	.075N
		*16(FZ10)	.075N
		*17(FX11)	.265N
		*18(FY11)	.135N
		*19(FZ11)	.089N
		*20(FY12)	.135N
		*21(FZ12)	.089N

*indicates specification violation ($E_{\omega u_i}^2 > \mu_i^2$)

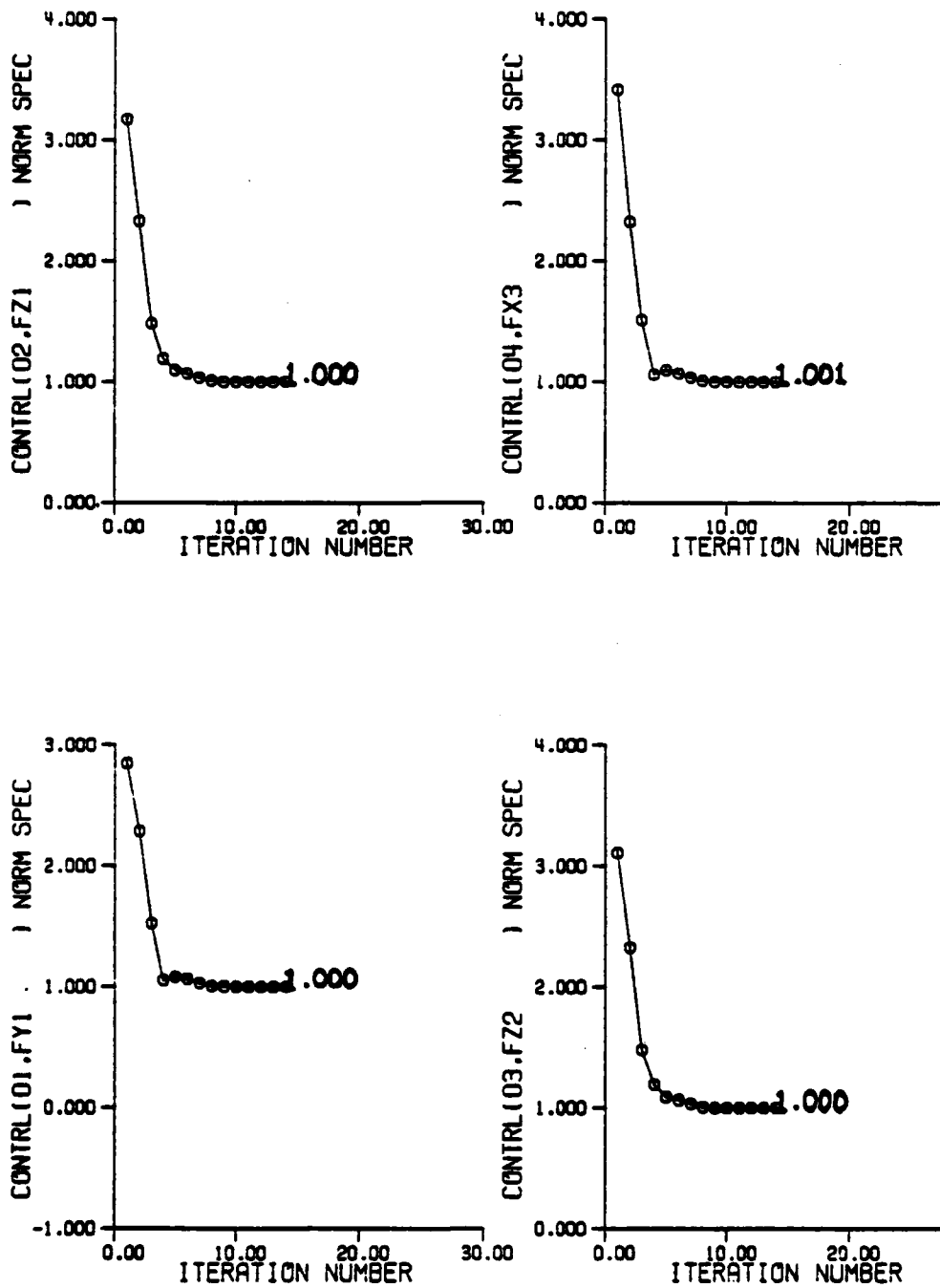


Figure 6.15: Telescope Input-Constrained Solution (actuators 1-4)

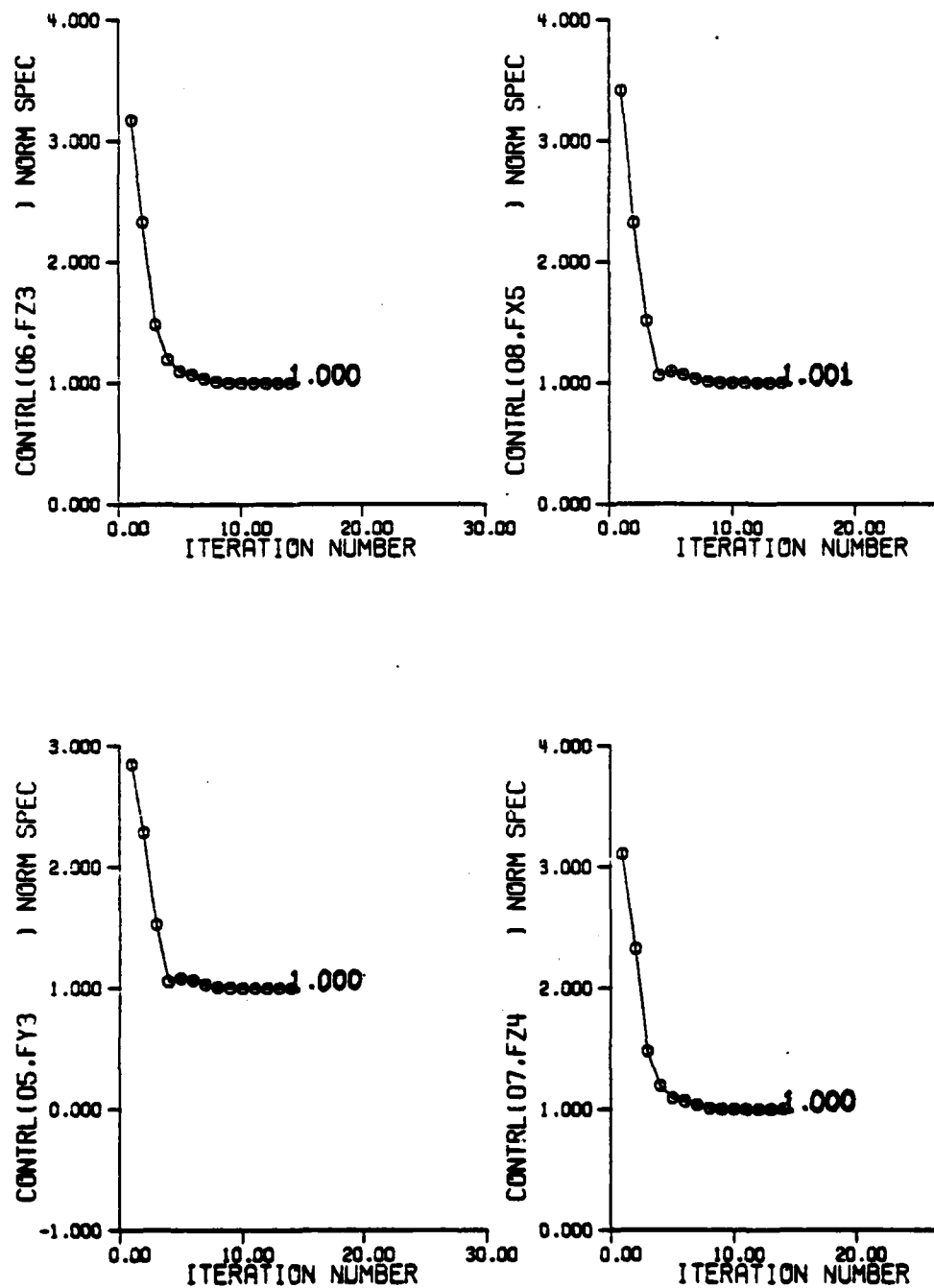


Figure 6.16: Telescope Input-Constrained Solution
(actuators 5-8)

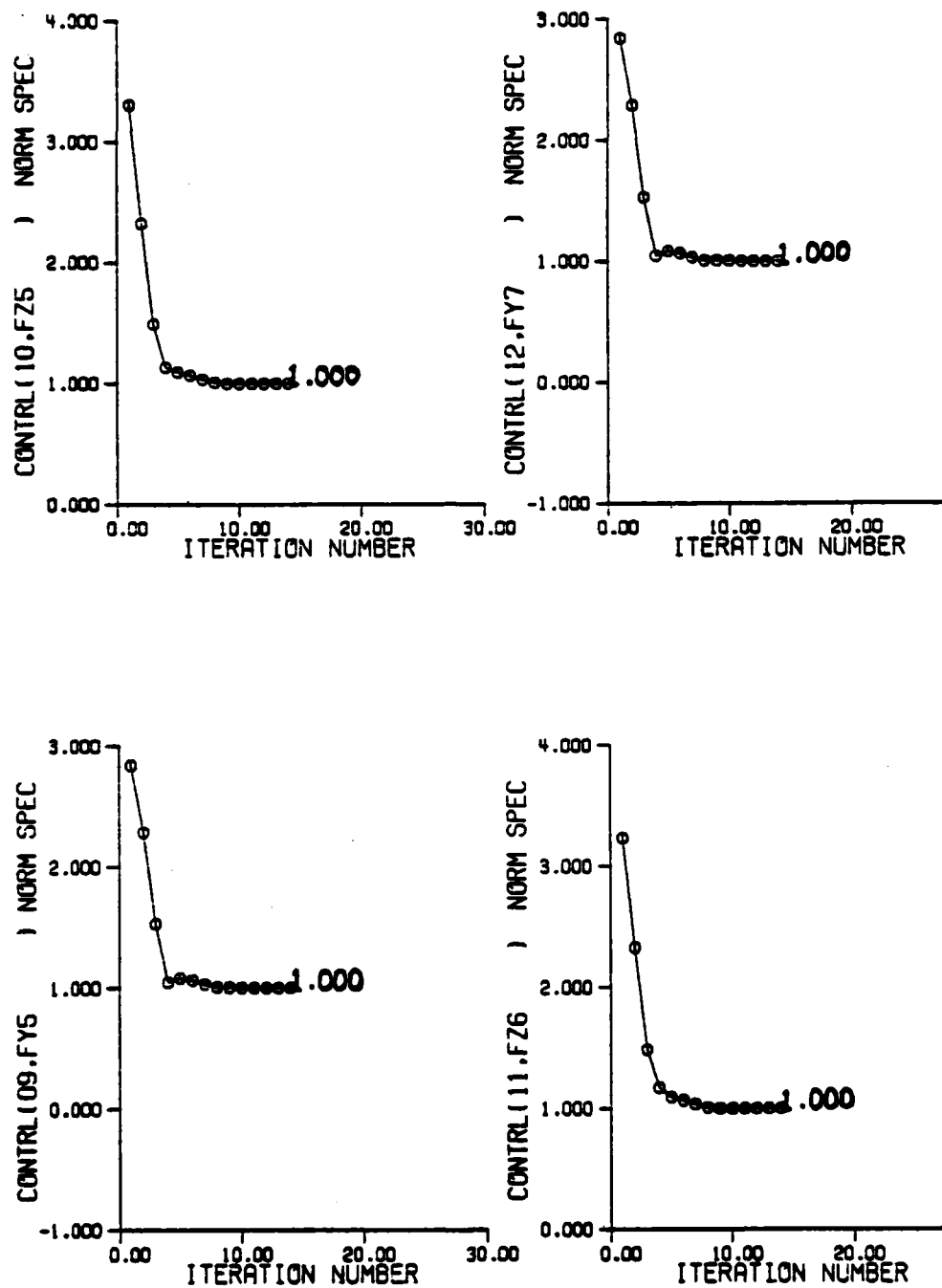


Figure 6.17: Telescope Input-Constrained Solution (actuators 9-12)

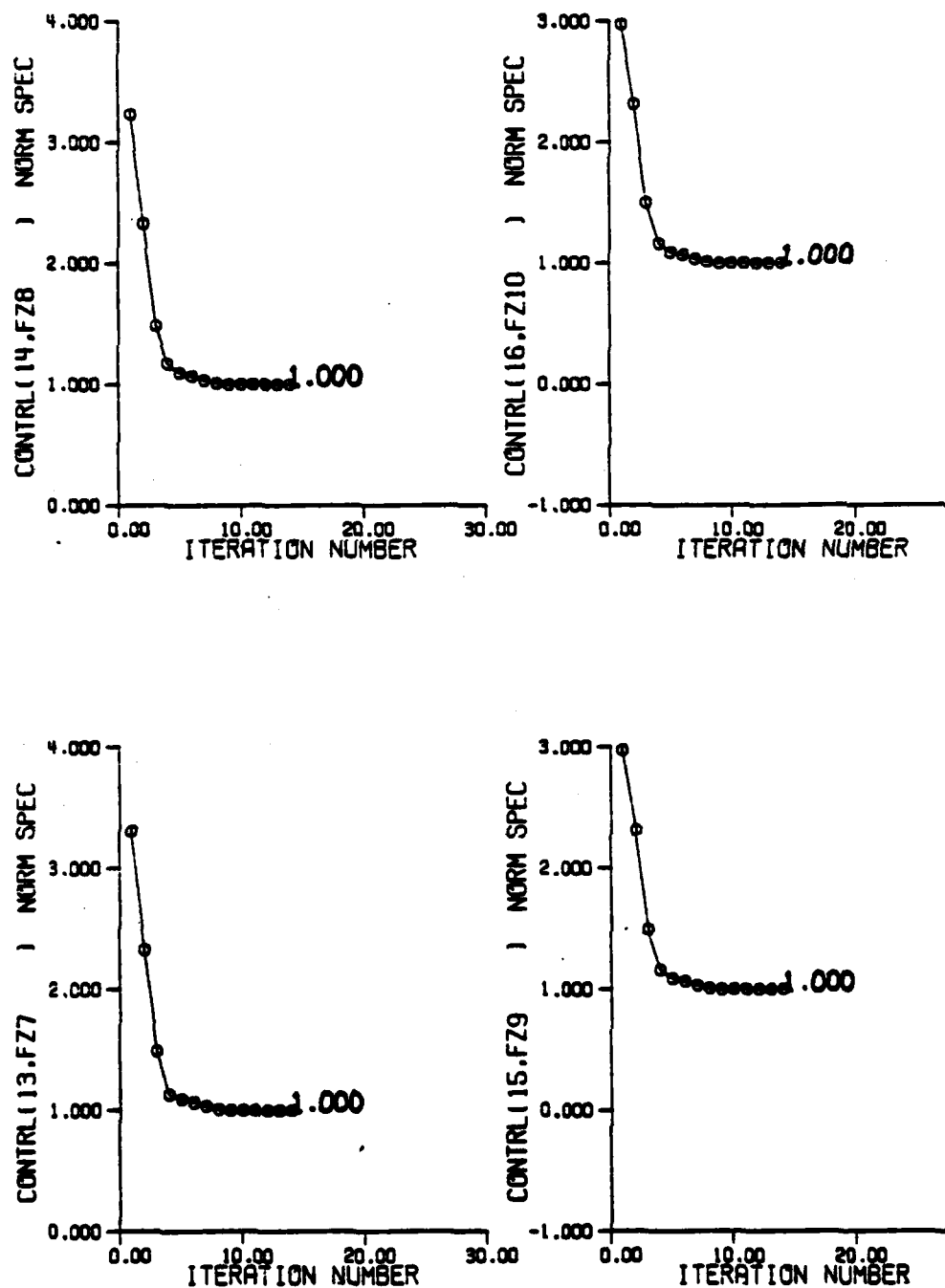


Figure 6.18: Telescope Input-Constrained Solution (actuators 13-16)

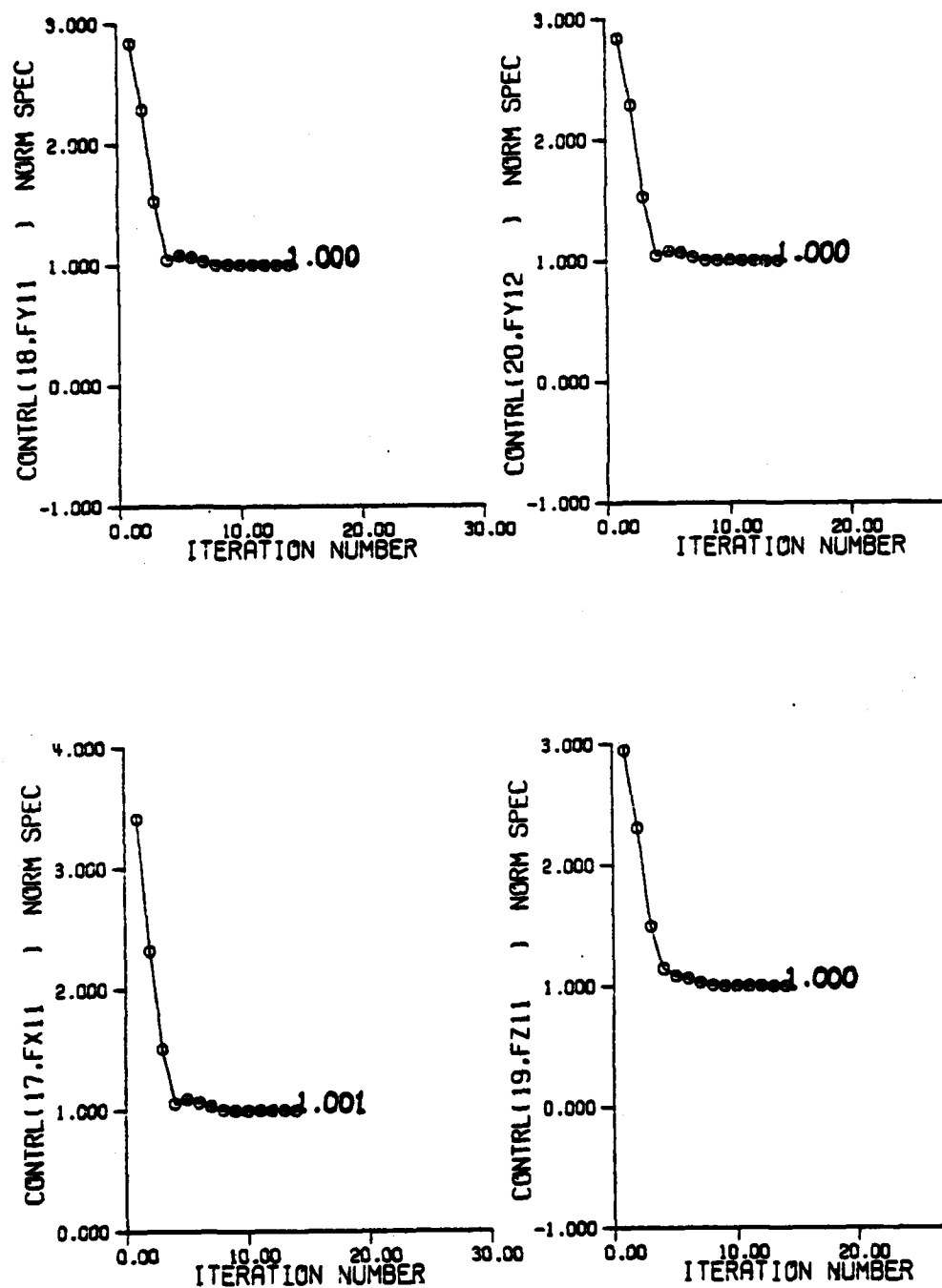


Figure 6.19: Telescope Input-Constrained Solution (actuators 17-20)

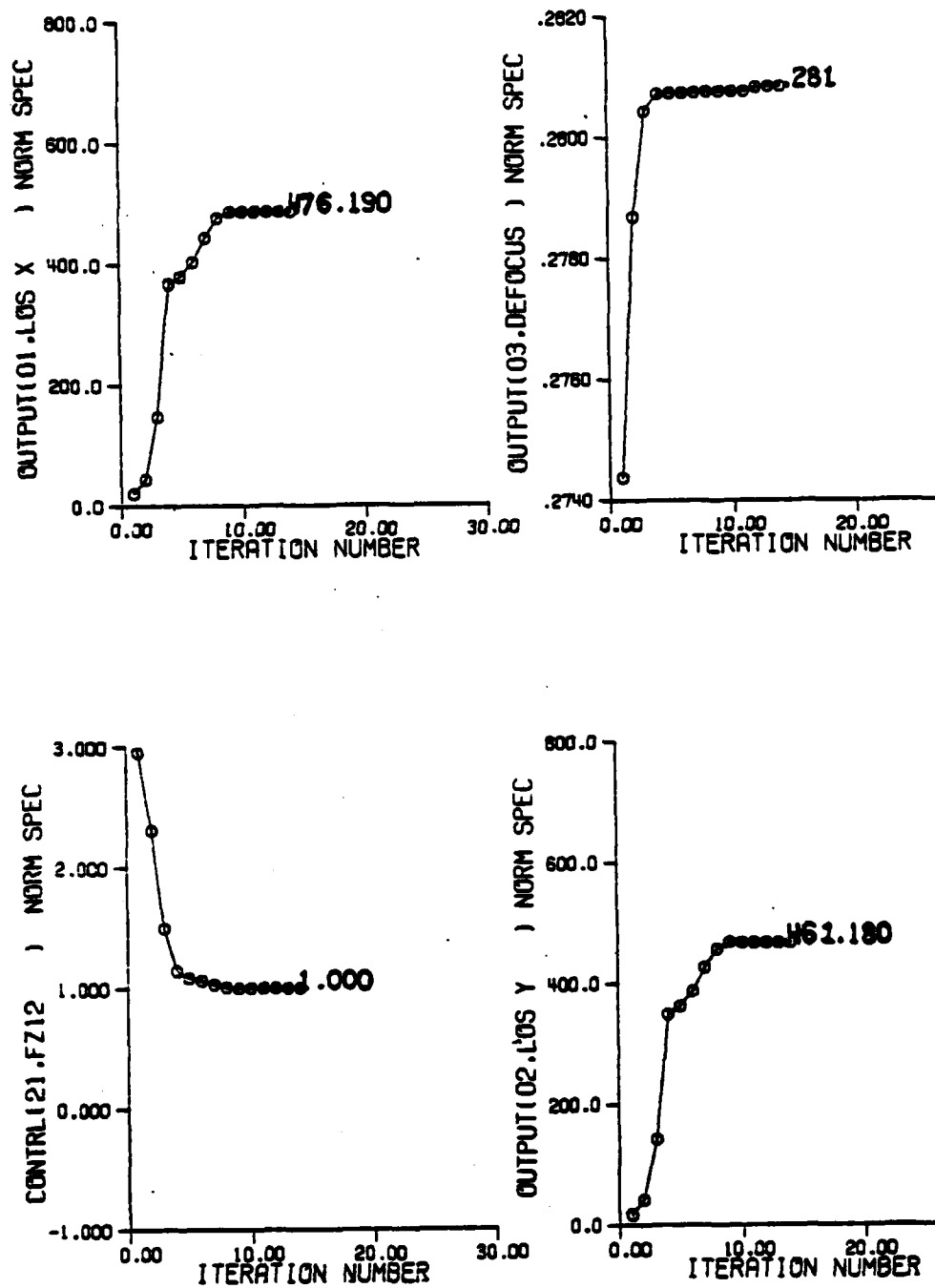


Figure 6.20: Telescope Input-Constrained Solution
(actuator 21, outputs 1-3)

The output-constrained search drove the outputs $y_i \in Y_c$ to their specification (i.e. LOS X, LOS Y) and zeroed the weight on the defocus output. The actuator specifications shown in Table 6.4 represent the minimum achievable input specifications required by 6.2b and promised by Theorem 5. These results were generated in 10 iterations of the algorithm and two 'step size resets' were required. The plots of Figure 6.21 show the *normalized* values of the outputs and a representative control versus iteration number.

6.6.3 Summary of Telescope Results

As in the case of the hoop column, LQGWTS converged to both the input and output constrained solutions for the telescope. There was a noticable difference in the convergence speed for the input and output constrained solutions, and this is a direct result of the fact that the input constrained option was simultaneously solving 22 transcendental equations while the output constrained case was solving only 3. In addition, the data indicated that an ϵ larger than .001 could have been used, and faster convergence would have resulted. In terms of 'reset iterations,' the algorithm lost only 2 out of the total 28 iterations performed on the telescope. Therefore, both the telescope and the hoop examples have demonstrated that LQGWTS can successfully tailor the LQG cost functional to satisfy the input and output constrained requirements of the SASLQG problem and this ability is *fundamental* to achieving a solution.

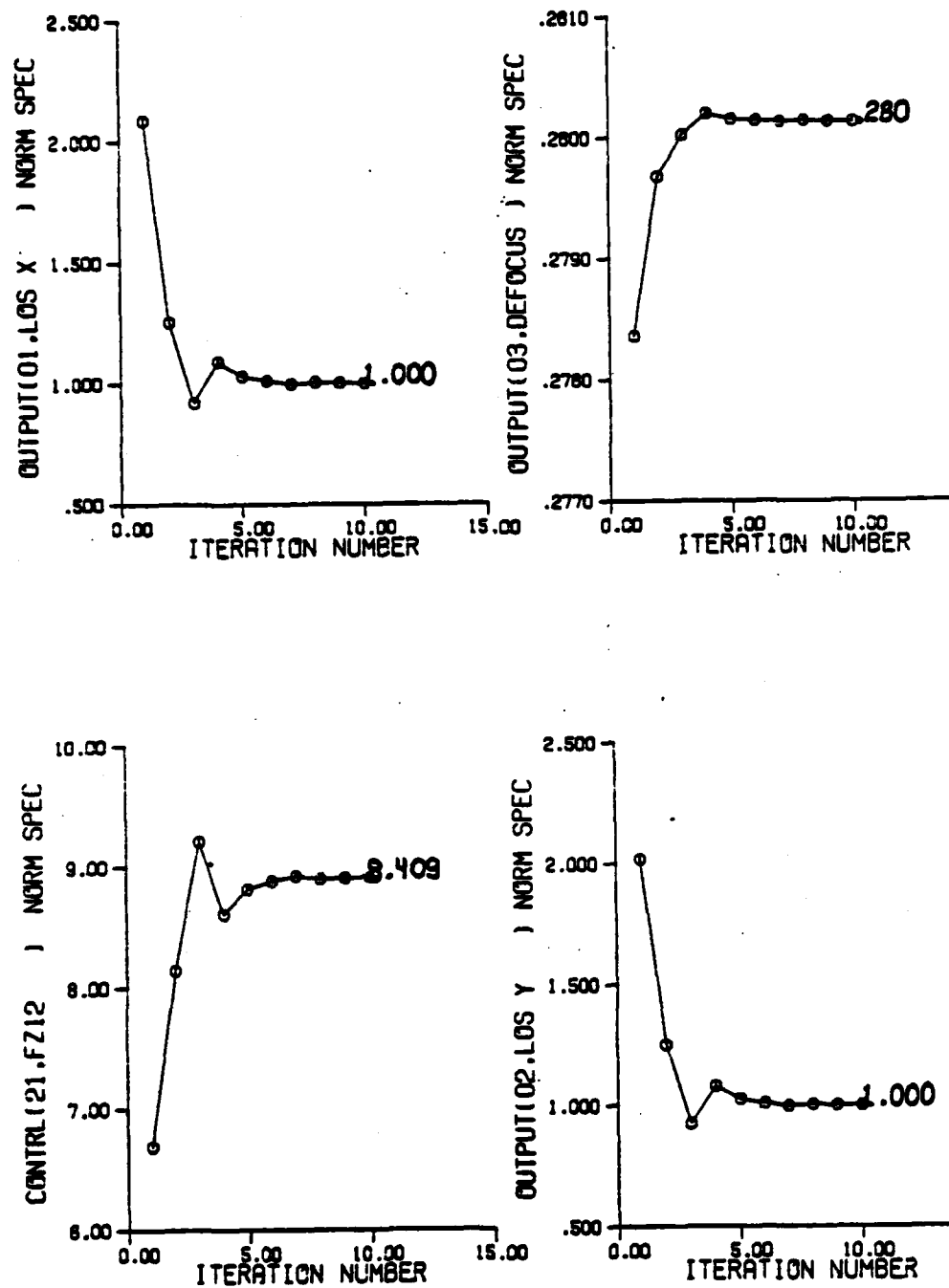


Figure 6.21: Telescope Output-Constrained Solution (actuator 21, outputs 1-3)

7.0 THE SASLQG ALGORITHM

As an introduction, a restatement of the SASLQG problem from Chapter 3 is appropriate.

SASLQG Problem Statement

Given: A system of type $S(n,k,m,\ell)$ which has only \bar{m} out of m actuators and $\bar{\ell}$ out of ℓ sensors available for the design of a steady state LQG regulator which must achieve (σ^2, μ^2) .

Required: Specify the closed-loop system which satisfies the following input-constrained or output-constrained requirements:

Input-constrained

If (σ^2, μ^2) are achievable,

$$(7.1a) \quad \min_S \sum_{i=1}^k E_{\omega} y_i^2 / \sigma_i^2 \quad \text{subject to} \quad E_{\omega} u_i^2 = \bar{\mu}_i^2 \quad \forall i = 1, \dots, \bar{m}$$

else,

$$(7.1b) \quad \min_S \sum_{i=1}^k E_{\omega} y_i^2 / \sigma_i^2 \quad \forall i: E_{\omega} y_i^2 > \sigma_i^2$$

$$\text{Subject to } E_{\omega} u_i^2 = \bar{\mu}_i^2 \quad \forall i = 1, \dots, \bar{m}$$

Output-constrained

If (σ^2, μ^2) are achievable,

$$(7.2a) \quad \text{Min}_S \sum_{i=1}^{\bar{m}} E_{\omega} u_i^2 / \bar{\mu}^2 \quad \text{subject to} \quad E_{\omega} y_i^2 = \sigma_i^2 \quad \forall i = 1, \dots, k$$

else,

$$(7.2b) \quad \text{Min}_S \sum_{i=1}^{\bar{m}} E_{\omega} u_i^2 / \bar{\mu}_i^2 \quad \forall i: E_{\omega} u_i^2 > \bar{\mu}_i^2$$

$$\text{Subject to } E_{\omega} y_i^2 = \sigma_i^2 \quad \forall i = 1, \dots, k$$

For the approach chosen by this research, there are, of course, two key elements in solving the SASLQG problem: First, tailoring the cost functional V so that the LQG controller which minimizes V also achieves the input-constrained requirements or the output constrained requirements, and second, evaluating the effectiveness of the role of each sensor and actuator in the minimization of this tailored V . Chapter 6 provides the algorithm LQGWTS for tailoring V so that either the input-constrained or the output-constrained requirements are satisfied by the LQG controller. Chapter 5 develops the sensor and actuator values v_i^{sen} and v_i^{act} which provide a relative measure of the effectiveness of each sensor and actuator in the minimization effort. The objective of this chapter is to combine LQGWTS and v_i^{act} , v_i^{sen} into an algorithm which solves the SASLQG problem. Section 7.1 presents the facts which were considered in developing the algorithm (SASLQG) along with a discussion of the general flow of the algorithm. While Section 7.2

presents the detailed steps of SASLQG. The algorithm is applied to the hoop-column and telescope models in Sections 7.3 and 7.4 respectively and concluding comments appear in Section 7.5.

7.1 Important Facts and Basic Algorithm Flow

The following facts merit consideration in combining LQGWS and v_i^{act} , v_i^{sen} to solve the SASLQG problem.

Facts

- (1) The data of Chapter 5 indicates that relative v_i^{act} (v_i^{sen}) ranking between actuators (sensors) can change as a function of Q and R. This fact is further illustrated by the results of [12] that showed the actuator location on a simple beam which minimized V was a function of Q and R.
- (2) Q_M , R_M , Q_M^* , R_M^* change when the actuator and/or sensor structure change. (i.e. the expressions $E_{\omega} y_i^2$, $E_{\omega} u_i^2$ are explicit and implicit functions of B and M which contain actuator and sensor configuration information).
- (3) The results of [11], [15] and [16] have shown the sensor and actuator selection problems to be highly coupled and that simultaneous selection works better than sequential selection. For a simple example of coupling note:

$$v_i^{\text{act}} = r_i^{-2} b_i^T \hat{K} \hat{X} K b_i - [B^T (K+L) B \omega^2]_{ii}$$

$$v_i^{\text{sen}} = v_{ii}^{-1} m_i^T P L P m_i$$

are both implicit and explicit functions of actuator and sensor terms.

- (4) v_i^{sen} and v_i^{act} do not provide insight into when deletion of a sensor or actuator will cause loss of measurability (observability) or controllability.
- (5) From Theorem 3 it is known that deleting sensors can *never* improve regulator performance.
- (6) From Theorem 2 it is known that deleting noisy sensors may improve regulator performance.

The preceding 6 facts suggest the following flow for the algorithm SASLQG.

SASLQG General Flow

- ① Run LQGWS (Input-constrained or output-constrained option) assuming that all admissible sensors and actuators are available and operating (i.e. $S(n,k,m,l)$)
- ② Determine v_i^{act} and v_i^{sen} for all sensors and actuators and then rank them from highest to lowest.
- ③ Throw out the lowest ranking sensor and actuator, and all others of 'nearly' the same ranking, if measurability and/or controllability of the system are not disturbed.
- ④ Return to ① until the appropriate number of actuators and sensors have been deleted (i.e. \bar{m} , \bar{l}). However, in light of Theorem 2, fewer actuators than \bar{m} might be desirable and actuator deletion should continue until the "minimum" achievable specifications (i.e. (7.1) or (7.2) are no longer improving.

Facts (1) and (2) have dictated the order of steps ① and ② in the above described general flow. More specifically, since the ranking of effectiveness values can vary as a function of Q and R , and we are interested in effectiveness rankings for a V which produces a controller that satisfies (7.1) or (7.2), the effectiveness values

should be based on the Q and R that specify the desired V .

Step ③ of the general flow results from facts (3) and (4).

Obviously caution must be exercised in deleting sensors and actuators with 'nearly' the same effectiveness values since facts (1) and (2) indicate that any perturbation in the sensor and actuator configuration can alter relative effectiveness rankings of the sensors and/or actuators. Also, the requirement for checking measurability and controllability is severe however, as a result of the work of Skelton and Hughes in [65], the controllability and observability of systems in modal form (i.e. (4.15)) can be done virtually by inspection for systems of any dimension.

Facts (5) and (6) influence the rationale behind step ④ of the general flow. If \bar{m} sensors are allowed then according to Theorem 3, \bar{m} sensors should be used, but, of course, Theorem 2 invalidates this philosophy for actuators. Since either the input-constrained requirements or output-constrained requirements are sought, the requirement of step ④ to check the improvement of these specifications can be met easily, by checking the appropriate sum at each iteration:

Input-constrained Solution

For (7.1a) check for decrease in:

$$(7.3) \quad \sum_{i=1}^k E_{\omega} y_i^2 / \sigma_i^2$$

For (7.1b) check for decrease in:

$$(7.4) \quad \sum_{i=1}^k E_{\omega} y_i^2 / \sigma_i^2, \quad \forall i: E_{\omega} y_i^2 > \sigma_i$$

Output Constrained Solution (for m_a actuators present)

For (7.2a) check for decrease in:

$$(7.5) \quad \sum_{i=1}^{m_a} E_{\infty} u_i^2 / \mu_i^2$$

or

$$1/m_a \sum_{i=1}^{m_a} E_{\infty} u_i^2 / \mu_i^2$$

For (7.2b) check for decrease in:

$$(7.6) \quad \forall i: E_{\infty} u_i^2 > \mu_i^2; \sum_{i=1}^{m_a} E_{\infty} u_i^2 / \mu_i^2 \quad \text{or} \quad 1/m_a \sum_{i=1}^{m_a} E_{\infty} u_i^2 / \mu_i^2$$

The average power ($1/m_a \sum_{i=1}^{m_a} E_{\infty} u_i^2 / \mu_i^2$) is considered by the output-constrained checks because the average power might be of more interest, and a decrease in total normalized power (i.e. $\sum_{i=1}^{m_a} E_{\infty} u_i^2 / \mu_i^2$) cannot guarantee a decrease in average power since m_a is also decreasing.

Before presenting the detailed algorithm the question raised in Chapter 5 concerning v_i^{act} and constrained input power must be addressed. The condition under which v_i^{act} was derived and tested in Chapter 5 assumed, implicitly, that control power was not limited to an extent which would prevent other actuators from being able to increase power to compensate for the essential (noiseless) contribution of a deleted actuator. In the input constrained situation, these conditions could certainly occur and did for the hoop column example. It was therefore necessary to modify v_i^{act} for finding SASLQG input-constrained specifications. After considerable testing, the following v_i^{act}

modification was found to work for *both* the hoop and the telescope examples:

$$(7.7) \quad v_i^{\text{act}_I} \triangleq \frac{v_i^u}{v_i^{w^a}} \left(\frac{v^u}{v^u - v_i^u} \right)$$

$$\text{where } v^u = \sum_{i=1}^m v_i^u$$

The expression in (7.7) does have intuitive appeal since it represents the product of an signal to noise ratio $\left(\frac{v_i^u}{v_i^{w^a}} \right)$ and a term $\left(\frac{v^u}{v^u - v_i^u} \right)$ which is the inverse of the fractional load carried by the remaining actuators. It should once more be re-emphasized that (7.7) was found to be a suitable measure of actuator effectiveness *only* in the input-constrained situation.

7.2 The Algorithm

The specific details of the algorithm SASLQG are presented in this section. The specific steps of the algorithm will be presented first and a brief discussion will follow.

Algorithm SASLQG

- ① Specify the number of sensors (\bar{l}) and actuators (\bar{m}) that can be used in the design. Choose the input-constrained or output-constrained option and run LQGWTS with all admissible sensors and actuators present.
- ② With the resulting LQG controller (i.e. (3.2) and (3.6)), calculate $v_i^{\text{sen}} = v_i^v$ for each sensor using (5.4f,g).

Input-constrained option:

Calculate $v_i^{\text{act}_I}$ for each actuator using (7.7) and (5.4a,d).

Output-constrained option:

Calculate v_i^{act} for each actuator using (5.8) and (5.4a,d).

- ③ Rank the sensors from highest to lowest based on v_i^{sen} values. Rank the actuators from highest to lowest based on $v_i^{\text{act}_I}$ or v_i^{act} values.
- ④ From the rankings in ③, identify the lowest ranking sensor(s) for which deletion would not disturb the measurability of the system. Also, identify the lowest ranking actuator(s) (from the ranking in ③) for which deletion would not disturb the controllability of the system.

- ⑤ Calculate the following value:

Input-constrained option

For condition 7.1a: calculate (7.3)

For condition 7.1b: calculate (7.4)

Output-constrained option

For condition 7.2a: calculate desired value in (7.5)

For condition 7.2b: calculate desired value in (7.6)

If the value is greater than the value for the previous iteration and the number of actuators is less than \bar{m} , restore the previous iteration actuator configuration and make no more actuator deletions.

- ⑥ If no sensor or actuator deletions are required. Stop.
- ⑦ Delete the identified sensors (i.e. drop the appropriate rows of M and V, and the appropriate columns of V). Delete the identified actuators (i.e. drop the appropriate columns of B,D,R,W and rows of R,W).

- 8.) Rerun the appropriate option of LQGWS for the reduced system and go to 2).

The fortran IV subroutine LQGWS has been modified to also calculate and rank the actuator and sensor effectiveness values (v_i^{act} or $v_i^{act_I}$) and v_i^v , and as mentioned in Chapter 6, Appendix E contains an explanation and listing of this program. Therefore, the first 3 steps of SASLQG have been automated. Steps 4), 5), and 6) are performed by the controls engineer based on the output of LQGWS and of course some form of controllability and measurability check. Both the hoop column and telescope models are in modal form and the observability checks developed by Skelton and Hughes [65] can be applied by inspection to these models. For systems not in modal form a transformation to modal form could be helpful. The steps 7) and 8) have been automated for the hoop column and telescope models by writing a calling program for the subroutine LQGWS. The calling program can delete any necessary row or column of the system matrices by simply specifying the actuator or sensor number to be deleted. These calling programs are labeled LQHOO and LQTELE and are also listed in Appendix E.

7.3 Hoop Column SASLQG Example

The SASLQG problem for the hoop column antenna model (S_{Hoop} (26, 24, 12, 39)) was posed in Section 4.3.2 as the H.SAS problem and is repeated here for continuity.

H.SAS PROBLEM

Given: S_{Hoop} (26, 24, 12, 39) with only 6 actuators and 12 sensors available for designing an LQG regulator

to achieve the (σ^2, μ^2) specifications of (4.16).

Required: Specify the closed-loop system which satisfies either (the input constrained requirements of (7.1) or the output constrained requirements of (7.2).

Before presenting the results, the following table of sensor labels is added to the output and actuator labels used in Tables 6.1 and 6.2. Note that ARX2 stands for an angular rate sensor in the X direction at node 2.

Table 7.1: Hoop Column Sensor Labels

Sensor Number	Label	Sensor Number	Label	Sensor Number	Label
1	AX2	14	AY10	27	Z119-Z10
2	AY2	15	AZ10	28	ARX2
3	AZ2	16	X101-X10	29	ARY2
4	X6-X2	17	Y101-Y10	30	ARZ2
5	Y6-Y2	18	Z101-Z10	31	ARX6
6	Z6-Z2	19	X107-X10	32	ARY6
7	X9-X2	20	Y107-Y10	33	ARZ6
8	Y9-Y2	21	Z107-Z10	34	ARX9
9	Z9-Z2	22	X113-X10	35	ARY9
10	X10-X2	23	Y113-Y10	36	ARZ9
11	Y10-Y2	24	Z113-Z10	37	ARX10
12	Z10-Z2	25	X119-X10	38	ARY10
13	AX10	26	Y119-Y10	39	ARZ10

7.3.1 Input-Constrained and Output-Constrained SASLQG Results

Tables 7.2 and 7.3 present the iteration by iteration results of the SASLQG input constrained and output constrained options applied to the hoop column antenna. The results indicated that no LQG controller existed to meet the (σ^2, μ^2) of (4.16). Therefore, the conditions

of (7.1b) and (7.2b) were sought. Both tables show the iteration number and the sensors and actuators identified for deletion in the next iteration with their effectiveness values in parentheses. Table 7.2 lists the normalized output sum (7.4), while Table 7.3 lists the average normalized control power (7.6). The last column in both tables is the number of sensors and actuators in the current design.

As might be expected for an input constrained case, the iteration data of Table 7.2 is more sensitive to actuator deletion than sensor deletion. Of particular interest is the large jump in the output value between iteration 6 and 7. For certain design situations this large shift could warrant a hardware change from 6 to 7 actuators. The (0) effectiveness values for (Z6-Z2), (Z9-Z2), and (Z10-Z2) result from the fact that these measurements correspond to rows in the M matrix which have very close to zero magnitude. If the longitudinal axis of the hoop-column was assumed incompressible in the NASTRAN development (a reasonable assumption) then that would account for the non-measurability of these sensor locations. As in the input-constrained case, the data of Table 7.3 was more sensitive to actuator deletion than sensor deletion. However, a significantly different sensor and actuator configuration was identified. The largest shift in the input value occurred between iterations 5 and 6 which, as in the input case, corresponds to a drop from 7 to 6 actuators. It should also be noted that if the sum of input power had been adopted as an effectiveness criterion instead of the average, the algorithm would have continued to delete actuators past 6 (i.e. \bar{m}).

Table 7.2: Hoop Column Input-Constrained SASLQG Results

Iteration Number	Identified Sensors ($v_{i, \text{sen}}$)	Actuators (v_{i, act_I})	Output Value (7.4)	Number of Sensors/Actuators
1	AZ10(.000253) AZ2 (.000243) Z6-Z2(0) Z9-Z2(0) Z10-Z2(0)	TX10(.69642)	300.15	39/12
2	Z119-Z10(.00840) AY2 (.00757) Z113-Z10(.00707)	TY10(.69671)	323.24	34/11
3	AX10(.0160) AY10(.0159) AX2(.0107)	TX9(.75236)	346.46	31/10
4	Y6-Y2(.114) Z107-Z10(.0768) X6-X2(.0682) Z101-Z10(.0596)	TY9(.74576)	410.39	28/9
5	ARZ9(.260) ARZ10(.258) ARZ6(.258)	TZ9(.84762)	472.46	24/8
6	Y9-Y2(.772) X9-X2(.768)	TY2(.82705)	639.97	21/7
7	Y119-Y10(1.242) Y107-Y10(1.238) Y101-Y10(1.236)	----	970.16	19/6
8	ARX6(2.037) ARX2(2.037)	----	975.48	16/6
9	ARY2(2.409) ARY6(2.408)	----	980.27	14/6
10	----	----	985.45	12/6

Table 7.3: Hoop Column Output-Constrained SASLQG Results

Iteration Number	Identified Sensors (v_1^{sen})	Identified Actuators (v_1^{act})	Ave Input Value (7.6)	Number of Sensors/Actuators
1	AZ10(.0004116) AZ2(.000397) Z6-Z2(0) Z9-Z2(0) Z10-Z2(0)	TZ10(-1.362) TZ9(-1.369)	3.275	39/12
2	AY1(.003362) AX10(.003358) AY2(.00226) AX2(.00226) Z113-Z10(.001942) Z119-Z10(.001884)	TZ6(-2.1405)	3.592	34/10
3	X6-X2(.01457) Y6-Y2(.01455) Z101-Z10(.0110) Z107-Z10(.0108)	TX10(-1.2055)	3.699	28/9
4	ARZ2(.02844) ARZ10(.02232) ARZ6(.02238)	TX9(-1.2917)	3.997	24/8
5	X9-X2(.0986) Y9-Y2(.0839)	TX6(-1.4793)	4.377	21/7
6	ARX6(.07648) ARX2(.07648)	----	4.829	19/6
7	Y107-Y10(.13395) ARY9(.1098)	----	4.857	17/6
8	X119-X10(.1557) X113-X10(.1555) X101-X10(.1551)	----	4.905	15/6
9	----	----	5.021	12/6

7.3.2 Sensor and Actuator Configuration Comparisons

It should be remembered that the goal of the SASLQG input-constrained and output-constrained algorithms is to locate the \bar{l} out of l sensors and \bar{m} out of m actuators such that (7.1) or (7.2) are best achieved. Since no proof of optimality is currently available for the SASLQG algorithm just presented, the important question is whether another configuration exists which can do better either in the input-constrained or output-constrained case. For the hoop-column, numerous configurations of 6 actuators and 12 sensors have been tested, and *none* have bettered the configurations defined by Tables 7.2 and 7.3. Of course, all possible configurations would have to be tested to theoretically verify the optimality of the configurations. Table 7.4 and 7.5 compare the input-constrained and output-constrained configurations respectively with each other and 2 other possible configurations. In case (2) of Table 7.4 TX2 of the input-constrained configuration was changed to TZ9. This gave the configuration of case (2) the use of all z torquers, the rational being that if 3 Z torquers did well, four would do better. For case (3) of Table 7.4 a form of collocation was attempted where all torquers were collocated with rate sensors. In case (4), the optimal SASLQG output-constrained configuration was tried and obviously doesn't do very well for an input constrained solution.

In the data of Table 7.5, case (2) is a collocation configuration similar to the one in Table 7.4. The measurements for case (3) of Table 7.5 were deliberately chosen so that measurability was lost with relatively disastrous results. Finally, in case (4) the

Table 7.4: Input-Constrained Configuration Comparison

Case Number Location Method:	(1) Input-Constrained	(2) All Z Torquers	(3) Collocation	(4) Output-Constrained
Actuators	TX2	TZ2	TX2	TX2
	TZ2	TX6	TY2	TY2
	TX6	TY6	TZ2	TZ2
	TY6	TZ6	TX6	TY6
	TZ6	TZ9	TY6	TY9
	TZ10	TZ10	TZ6	TY10
Sensors	X10-X2	X10-X2	X10-X2	X10-X2
	Y10-Y2	Y10-Y2	Y10-Y2	Y10-Y2
	X101-X10	X101-X10	X119-X10	Y101-Y10
	X107-X10	X107-X10	ARX2	X107-X10
	X113-X10	X113-X10	ARY2	Y113-Y10
	Y113-Y10	Y113-Y10	ARZ2	Y119-Y10
	X119-X10	X119-X10	ARX6	ARY2
	ARZ2	ARZ2	ARY6	ARY6
	ARX9	ARX9	ARZ6	ARY9
	ARY9	ARY9	ARX10	ARZ9
	ARX10	ARX10	ARY10	ARX10
	ARY10	ARY10	ARZ10	ARY10
	985.45*	1050.39	1118.79	3628.11
	(7.1b) Specification *=> minimum			

Table 7.5: Output-Constrained Configuration Comparison

Case Number Location Method:	(1) Output-Constrained	(2) Collocation	(3) Detectability	(4) Input-Constrained
Actuators				
	TX2	TX2	TX2	TX2
	TY2	TY2	TY2	TZ2
	TZ2	TZ2	TZ2	TX6
	TY6	TX6	TX6	TY6
	TY9	TY6	TY6	TZ6
	TY10	TZ6	TZ6	TZ10
Sensors				
	X10-X2	X10-X2	X10-X2	X10-X2
	Y10-Y2	Y10-Y2	Y10-Y2	Y10-Y2
	Y101-Y10	Y113-X10	X113-X10	X101-X10
	X107-X10	ARX2	X119-X10	X107-X10
	Y113-Y10	ARY2	ARX2	X113-X10
	Y119-Y10	ARZ2	ARY2	Y113-Y10
	ARY2	ARX6	ARX6	X119-X10
	ARY6	ARY6	ARY6	ARZ2
	ARY9	ARZ6	ARX9	ARX9
	ARZ9	ARX10	ARY9	ARY9
	ARX10	ARY10	ARX10	ARX10
	ARY10	ARZ10	ARY10	ARY10
(7.2b) Average Specification \rightarrow minimum	5.0205*	5.463	14.389	5.235

optimal input-constrained SASLQG solution is seen to be a better approximation to an output-constrained solution than the converse condition in Table 7.4.

7.3.3 Physical Insights

Tables 7.6 and 7.7 provide the minimum achievable SASLQG specifications for the input-constrained solution and output-constrained solution of the H.SAS problem. These specifications are valuable in the sense that they represent a *physically* realizable set of specifications for the given hoop model and sensor and actuator noise characteristics, and an LQG controller has already been designed that achieves these specifications.

The data presented in Tables 7.6 and 7.7 offer other useful insights into the control problem. As noted in Chapter 6, the output specifications of Table 7.6 represent the minimum achievable specifications when all available control power (in a stochastic sense) is being used. The outputs which remain the farthest above specification represent those outputs which, for the given specifications, are the hardest to regulate. Dividing the (σ^2, μ^2) specifications of (4.16) into the results of Table 7.6 the following outputs were found to be significantly above specification and are ordered from highest to lowest: 16, 11, 22, 14, 13, 19, 6, 3, 9, 20, 17, 12, 23, 10, 15. Combining the output labels of Table 6.1 with the hoop-column schematic of Figure 4.1 provides a physical interpretation for these outputs. Outputs 14, 16, 20, and 22 are tangential displacements of the hoop with respect to the feed horn at node 10. Outputs 13, 17, 19 and 23 represent radial displacements of the hoop with respect to the feed

Table 7.6: H.SAS Input-Constrained Specifications

Output #	$\sqrt{E_{y_i}^2}$ (minimum achievable)	Actuator #	Specifications
1 (AX2)	.534 $\widehat{\text{sec}}$	1 (TX2)	10.000 dn-cm
2 (AY2)	1.779 $\widehat{\text{sec}}$	2 (TZ2)	10.000 dn-cm
3 (AZ2)	1265.6 $\widehat{\text{sec}}$	3 (TX6)	10.000 dn-cm
4 (AX10-AX2)	.026 $\widehat{\text{sec}}$	4 (TY6)	10.000 dn-cm
5 (AY10-AY2)	.085 $\widehat{\text{sec}}$	5 (TZ6)	10.000 dn-cm
6 (AZ10)	1311.7 $\widehat{\text{sec}}$	6 (TZ10)	10.000 dn-cm
7 (X6-X2)	1.247 mm		
8 (Y6-Y2)	.374 mm		
9 (X9-X2)	8.161 mm		
10 (Y9-Y2)	2.450 mm		
11 (X10-X2)	18.984 mm		
12 (Y10-Y2)	5.700 mm		
13 (X101-X10)	13.671 mm		
14 (Y101-Y10)	15.141 mm		
15 (Z101-Z10)	.767 mm		
16 (X107-X10)	20.573 mm		
17 (Y107-Y10)	7.503 mm		
18 (Z107-Z10)	.445 mm		
19 (X113-X10)	13.519 mm		
20 (Y113-Y10)	7.677 mm		
21 (Z113-Z10)	.137 mm		
22 (X119-X10)	16.949 mm		
23 (Y119-Y10)	4.680 mm		
24 (Z119-Z10)	.080 mm		

Table 7.7: H.SAS Output-Constrained Specifications

Output #	$\sqrt{\epsilon_{\omega y_i}^2}$	Actuator #	$\sqrt{\epsilon_{\omega u_i}^2}$ (minimum achievable)
1 (AX2)	.015 $\overline{\text{sec}}$	1 TX2	72.91 dn-cm
2 (AY2)	.015 $\overline{\text{sec}}$	2 TY2	26.145 dn-cm
3 (AZ2)	11.588 $\overline{\text{sec}}$	3 TZ2	105.47 dn-cm
4 (AX10-AX2)	.001 $\overline{\text{sec}}$	4 TY6	26.138 dn-cm
5 (AY10-AY2)	.001 $\overline{\text{sec}}$	5 TY9	31.750 dn-cm
6 (AZ10)	12.000 $\overline{\text{sec}}$	6 TY10	38.812 dn-cm
7 (X6-X2)	.010 mm		
8 (Y6-Y2)	.010 mm		
9 (X9-X2)	.068 mm		
10 (Y9-Y2)	.068 mm		
11 (X10-X2)	.158 mm		
12 (Y10-Y2)	.158 mm		
13 (X101-X10)	.104 mm		
14 (Y101-Y10)	.158 mm		
15 (Z101-Z10)	.007 mm		
16 (X107-X10)	.158 mm		
17 (Y107-Y10)	.156 mm		
18 (Z107-Z10)	.008 mm		
19 (X113-X10)	.122 mm		
20 (Y113-Y10)	.158 mm		
21 (Z113-Z10)	.001 mm		
22 (X119-X10)	.158 mm		
23 (Y119-Y10)	.091 mm		
24 (Z119-Z10)	.001 mm		

horn. Twisting (i.e. z rotation) between the top and bottom of the column is represented by outputs 3 and 6, while bending and x/y rotation of the column are represented in outputs 11 and 12. Therefore, for the given sensor and actuator locations and actuator variance constraints, the most difficult control problems, respectively, are regulating the hoop x-y plane rotations with respect to the feed horn, keeping the hoop centered with respect to the feed horn, and minimizing the twisting, bending, and x/y rotation of the column. It should be mentioned that similar output information can also be obtained from the output constrained solution by noting which outputs require the largest q_i 's to achieve their specification.

Theorem 5 promises, that for the given output specifications, the torquer specifications in Table 7.7 represent the smallest possible *average* deviation from the required torquer specifications. A natural consequence of this situation is that the torquers in Table 7.7 with larger values of $E_{\omega} u_i^2$ are more critical to overall performance and therefore are candidates for the best hardware available (i.e. most reliable, least noisy etc). The data indicates that the z-torquer is the most critical actuator, and since hoop rotation with respect to the feed horn and column twisting are some of the most difficult outputs to control, this result is certainly not surprising. It should also be mentioned that similar critical input information can be obtained from the input-constrained solution by noting which inputs require the largest r_i 's to operate at their specification. Comparable insights and results exist for the telescope SASLQG example presented in the next section.

7.4 Telescope SASLQG Example

The SASLQG problem for the solar optical telescope model (S_{tele} (24, 3, 21, 45)) was posed in Section 4.3.2 as the T.SAS problem. The problem is repeated here again for continuity.

T.SAS Problem

Given: S_{tele} (24, 3, 21, 45) with only 12 actuators and 12 sensors available for designing an LQG regulator to achieve the (σ^2, μ^2) specification of Table 4.14.

Required: Specify the closed-loop system which satisfies either the input constrained requirements of (7.1) or the output constrained requirements of (7.2).

Before presenting the results, the following table of sensor labels is added to the output and actuator labels used in Tables 6.3 and 6.4: Note that Y1 represents a linear displacement sensor in the Y direction at node 1 and LRZ3 represents a linear rate sensor in the Z direction at node 3.

7.4.1 Input-Constrained and Output-Constrained Results

Tables 7.9 and 7.10 present the iteration by iteration results of the SASLQG input-constrained and output constrained options when applied to the T-SAS problem. The results indicated that no LQG controller existed to meet the (σ^2, μ^2) specifications of Table 4.14 and therefore the conditions of 7.1b and 7.2b were sought. The tables follow the same format as Tables 7.2 and 7.3.

The first point of interest in the data is that in iteration 1 of both solutions, the measurements of the outputs were deleted. It should be remembered, however, that the goal of Kalman-Bucy filter is

Table 7.8: Telescope Sensor Labels

Sensor Number	Label	Sensor Number	Label	Sensor Number	Label
1	LOSX	16	Z7	31	LRZ4
2	LOSX	17	Z8	32	LRX5
3	DEFOCUS	18	Z9	33	LRX5
4	Y1	19	Z10	34	LRZ5
5	Z1	20	X11	35	LRZ6
6	Z2	21	Y11	36	LRX7
7	X3	22	Z11	37	LRZ7
8	Y3	23	Y12	38	LRZ8
9	Z3	24	Z12	39	LRZ9
10	Z4	25	LRX1	40	LRZ10
11	X5	26	LRZ1	41	LRX11
12	Y5	27	LRZ2	42	LRX11
13	Z5	28	LRX3	43	LRZ11
14	Z6	29	LRX3	44	LRX12
15	Y7	30	LRZ3	45	LRZ12

Table 7.9: Telescope Input-Constrained SASLQG Results

Iteration Number	Identified Sensors (v_i^{sen})	Identified Actuators (v_i^{act})	Output Value (7.4)	Number of Sensors/Actuators
1	Y3(.01176) Y1(.01176) LOSX(.00003) LOSX(.000002) DEFOCUS(0)	FX11(.50204)	937.38	45/21
2	Y12(.04943) Y7(.04943) Y11(.04943) Y5(.04943)	FX5(.4600)	804.27	40/20
3	Z6(.050868) Z8(.050864) Z5(.04841) Z7(.04841)	FY12(.66234) FY7(.66234)	627.29	36/19
4	Z2(.07382) Z4(.07182) Z3(.06732) Z1(.06595)	FY11(.61612) FY5(.61612)	726.68	32/17
5	Z10(.13672) Z9(.13672)	FZ11(.6171) FZ12(.6171)	745.53	28/15
6	LRV3(.2820) LRV1(.2820) Z11(.2289) X3(.1920)	FZ9(.6037)	790.47	26/13
7	LRZ7(.3457) LRZ8(.3451) LRZ5(.3040)	----	788.48	22/12
8	LRZ4(.5234) LRZ3(.5173) LRZ1(.4515) LRZ6(.3895)	----	788.96	19/12
9	LRZ9(1.214) LRX3(.62263) LRZ2(.60428)	----	790.70	15/12
10	----	----	794.72	12/12

Table 7.10: Telescope Output-Constrained SASLQG Results

Iteration Number	Identified Sensor (V_i^{sen})	Identified Actuators (V_i^{act})	Ave. Input Value (7.6)	Number of Sensors/Actuators
1	Y3(.008197) Y1(.008195) LOSX(.000015) LOSX(.000002) DEFOCUS(0)	FX11(-7.980)	7.483	45/21
2	Z5(.02955) Z7(.02955)	FX5(-8.796)	7.393	40/20
3	Y12(.03848) Y7(.03848) Y11(.03848) Y5(.03848) Z2(.03835) Z4(.03835) Z1(.03570) Z3(.03570) Z6(.03460) Z8(.08460)	FY7(-4.003) FY12(-4.003) FY11(-4.003) FY5(-4.003)	7.221	38/19
4	Z10(.08602) Z9(.08602) X3(.08277)	FZ11(-4.8877) FZ12(-4.8877)	7.789	28/15
5	LRX3(1.0815) LRX1(1.0813)	FZ9(-5.414)	8.155	25/13
6	LRZ7(.12273) LRZ5(.11553)	----	8.339	23/12
7	LRZ1(.16805) Z11(.16129) LRZ6(.13838) LRZ8(.13615)	----	8.359	21/12
8	LRX3(.2237) LRZ2(.2191) LRZ4(.2077) LRZ3(.1960)	----	8.416	17/12
9	LRZ9(.47256)	----	8.492	13/12
10	----	----	8.535	12/12

to estimate the state of the system *not* necessarily the output. For the telescope example, the noise present on the output measurements made them unattractive to the filter. For Table 7.9, iterations 3 and 4 indicate that 19 or possibly 18 actuators are an optimal configuration if hardware constraints permit. From Table (7.10) the optimal number lies between 19 and 15. It should be noted, however, that the output value in Table 7.9 again began decreasing from iteration 6 to iteration 7, and according to the algorithm, actuator deletions should have continued. This oversight was just recently noted and time has not permitted a continuation of the actuator deletion sequence to determine if fewer than 12 actuator is a better configuration. As was the case for the hoop, iterations 7-10 in both tables indicate that the sensor deletions are not having as much impact on the regulator design as actuator deletions. Finally although not immediately obvious from the data in the tables, both the input-constrained and the output-constrained options converged to the *same* sensor and actuator configuration.

7.4.2 Sensor and Actuator Configuration Comparisons

As in the hoop example numerous other configurations were tested and compared to the SASLQG solution to the T.SAS problem. Since both the input-constrained and output-constrained options converged to the same configuration, the results of both comparisons are presented in Table 7.11 for the same test configurations. None of the tested configurations did better than the LQGSAS solution including of course the configurations in Table 7.11. Case (2) is a configuration which collocates the actuators with the majority of the optimal sensor locations defined in the SASLQG solution. Case (3) is a

Table 7.11: Input/Output-Constrained Configuration Comparison (T.SAS)

Case Number Location Method:	(1) Input-Constrained Output-Constrained	(2) Collocation, 1	(3) Collocation, 2	(4) Actuator in Version
Actuators	FY1 FZ1 FZ2 FX3 FY3 FZ3 FZ4 FZ5 FZ6 FZ7 FZ8 FZ10	FX5 FY5 FZ5 FY7 FZ7 FZ9 FZ10 FX11 FY11 FZ11 FY12 FZ12	FY1 FZ1 FZ2 FX3 FY3 FZ3 FZ4 FZ5 FZ6 FZ7 FZ8 FZ10	FZ2 FX3 FX5 FY5 FY7 FZ9 FZ10 FX11 FY11 FZ11 FY12 FZ12
Sensors	X5 X11 Y12 Z12 LRX5 LRY5 LRY7 LRZ10 LRX11 LRY11 LRZ11 LRY12 LRZ12	X11 Y12 LRX5 LRY5 LRY7 LRZ9 LRZ10 LRX11 LRY11 LRZ11 LRY12 LRZ12	X5 Y7 LRY1 LRZ1 LRX3 LRY3 LRZ3 LRX5 LRY5 LRZ5 LRY7 LRZ7	X11 Y12 LRX5 LRY5 LRY7 LRZ9 LRZ10 LRX11 LRY11 LRZ11 LRY12 LRZ12
(7.1b) Specification	794.72*	3676.20	810.67	3458.2
(7.2b) Specification *→ minimum	102.42*	152.868	109.896	154.116

collocation of the sensors with the optimal actuator locations of the SASLQG solution. Comparing cases (2) and (3) further indicates that actuator location is driving the control design process more than the sensors. The primary reason for this being the number of actuators, and the power constraints, and noise level on the actuators. Case (4) of Table (7.11) used all the actuator locations rejected by the SASLQG algorithm plus the majority of the optimal sensor locations.

7.4.3 Physical Insights

Tables 7.12 and 7.13 provide the minimum achievable specifications for the SASLQG input-constrained solution of the T.SAS problem. As in the hoop column example, these specifications are valuable in the sense that they represent a *physically* realizable set of specifications for the given telescope model ($S(24, 3, 12, 12)$), and an LQG controller has already been designed that achieves these specifications.

The data presented in Tables 7.12 and 7.13 also offer useful insights into the control problem. Remembering that the output specification on LOSX and LOSY is 65.2 sec, and that the specification or defocus is .001 mm, it is easy to see that LOSX and LOSY are well above specification. Since the specifications are the minimum achievable specifications, LOSX and LOSY must be the most difficult outputs to control. Using this same reasoning and recalling that the actuator specifications are .01N, the most critical actuator in the current control configuration is FZ10.

7.5 Concluding Comments

The purpose of this chapter has been to develop and test an algorithm which solves the SASLQG problem using the actuator and

Table 7.12: T.SAS Input-Constrained Specifications

Output #	$\sqrt{E_{\omega} y_i^2}$ (minimum achievable)	Actuator #	$\sqrt{E_{\omega} u_i^2}$ Specification
1(LOSX)	7.965°	1(FY1)	.01N
2(LOSX)	6.428°	2(FZ1)	"
3(DEFocus)	.00002 mm	3(FZ2)	"
		4(FX3)	"
		5(FY3)	"
		6(FZ3)	"
		7(FZ4)	"
		8(FZ5)	"
		9(FZ6)	"
		10(FZ7)	"
		11(FZ8)	"
		12(FZ10)	"

Table 7.13: T.SAS Output-Constrained Specifications

Output #	$\sqrt{E_{\omega} y_i^2}$	Actuator #	$\sqrt{E_{\omega} u_i^2}$ (Minimum achievable)
1(LOSX)	65.2 sec	1(FY1)	.059N
2(LOSX)	65.2 sec	2(FZ1)	.091N
3(DEFocus)	.0002 mm	3(FZ2)	.084N
		4(FX3)	.105N
		5(FY3)	.063N
		6(FZ3)	.070N
		7(FZ4)	.114N
		8(FZ5)	.060N
		9(FZ6)	.0585N
		10(FZ7)	.049N
		11(FZ8)	.075N
		12(FZ10)	.191N

sensor effectiveness values of Chapter 5 and the weight selection algorithm of Chapter 6. This was accomplished by identifying known facts about the SASLQG problem and then using these facts to develop an intuitively appealing algorithm which has no current proof of optimality. The algorithm was successfully applied to the substantial hoop column and solar optical telescope models. The documented results have supplied not only a sensor and actuator configuration but also insight into the optimal number of noisy actuators, and the most demanding outputs and critical actuators for the given control design. Finally, if the desired variance specifications cannot be met, the algorithm provides a set of minimum achievable specifications along with an LQG controller which produces them. With the results of this chapter, a summary of the entire research may now be presented.

8.0 CONCLUSION

This research has developed and tested an algorithm which aids the controls engineer in placing sensors and actuators (inputs) in a linear stochastic system $S(n,k,m,\ell)$ to 'best achieve' a set of variance specifications (σ^2, u^2) on the outputs and inputs of the system. The term 'best achieve' has been defined in the introduction to be the sensor and actuator configuration which enables a controller to do either of the following: Meet the input specifications while minimizing a sum of output variances normalized by their specification (σ^2) , (i.e. input-constrained solution), or meet the output specifications while minimizing a sum of input variances normalized by their specification (u^2) (i.e. output-constrained solution).

The approach taken to solve this sensor and actuator selection (SAS) problem was to use LQG theory to specify a structure for the controller, and then develop an algorithm (SASLQG) that places sensors and actuators in this controller structure to achieve either the input-constrained or output-constrained solution. The advantages and disadvantages to this approach were discussed in Chapter 3. The main advantage being the mathematical ease with which LQG theory addresses variance constraints, and the main disadvantage being that there may be other controller structures which do better.

8.1 Contributions

In applying LQG theory to solve the SAS problem two specific extensions of the theory resulted. The first was the development of the sensor and actuator effectiveness values v_i^{sen} and v_i^{act} in Chapter 5. These values determine the importance of each sensor and actuator to the LQG controller when *both* the sensors and actuators are assumed noisy. The second extension was the development of the Algorithm LQGWS in Chapter 6. This algorithm provides a systematic method for adjusting the weighting matrices in the LQG cost functional V so that the controller which minimizes V also satisfies either the input-constrained or output-constrained variance requirements.

These two extensions were combined to form the sensor and actuator selection algorithm SASLQG in Chapter 7. The algorithm was applied to two substantial models of large space structures and the resulting configurations although not guaranteed to be optimal achieved better performance than any alternative configuration tested. As noted in Chapter 7, the algorithm also provides insight into the sensitivity of the controller design to sensor and actuator deletions and therefore, insight into an optimal number for both sensors and actuators. Lastly, the algorithm also provides information which identifies the most demanding outputs and the critical actuators for the final sensor and actuator configuration.

8.2 Recommendations for Further Research

It is felt that the algorithm SASLQG has made an important contribution to the problem of selecting noisy sensors and actuators for regulating linear systems with variance constraints on both the inputs

and outputs of the system. In light of this contribution the following recommendations are made for further research:

8.2.1 Discrete Time Systems/Deterministic Systems

The results of this research should be extendable to both discrete time systems and deterministic systems. The only changes currently foreseen for a discrete time system would be using discrete versions of the continuous, steady state Riccati and Lyapunov equations currently used in the algorithm. For deterministic systems, the regulation constraints will take different forms but the process of tailoring the cost functional to achieve these constraints and then determining sensor and actuator contributions should still be possible using a modified version of the LQGWS update equations and v_i^{act} and v_i^{sen} . The specific details should be investigated.

8.2.2 Sensors Constraints

It is conceivable that variances constraints on the sensors of a system (i.e. $E_{\infty} z_i^2$) may need to be within some bound. Incorporation of these bounds appears possible through the techniques developed in this research and should be researched.

8.2.3 SASLQG Simplification

The algorithm SASLQG requires running the algorithm LQGWS at each iteration to tune the cost functional before the next set of deletions occur. Experience has shown that once the original tuning has been done (i.e. LQGWS run on the first iteration), the subsequent tunings of the cost functional may not be necessary. If conditions for this result could be determined a substantial computational burden of SASLQG could be removed.

AD-A135 442 SELECTION OF NOISY SENSORS AND ACTUATORS FOR REGULATION OF LINEAR SYSTEMS(U) AIR FORCE INST OF TECH
WRIGHT-PATTERSON AFB OH MIL-RESEARCH AUG 67 3/3

343

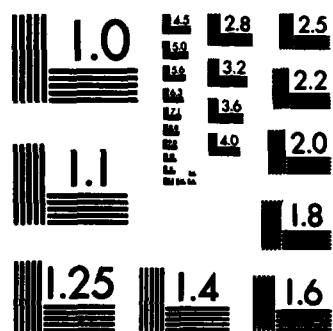
UNCLASSIFIED AFIT/CI/NR-83-59D F/G 12/1 NL

AFIT/CI/NR-83-59D

F/G 12/1

NL

84. 104
 104
 104



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

8.2.4 Non-Diagonal Q and R

While it is known that any cost functional which uses a non-diagonal Q and R can be converted to an equivalent diagonal Q and R cost functional by a unitary transformation on the system inputs and outputs. It is currently not clear what advantages (if any) are gained by using the off diagonal elements of the weighting matrices. This area needs to be investigated.

8.2.5 $PWR(j)$ and $PWR_y(j)$

There is certainly no guarantee that the $PWR(j)$ and $PWR_y(j)$ sequences chosen for the exponents in the LQGWTS update equations are the best sequences. Investigation could continue in this area by testing sequences such as e^j or $\frac{e^j}{j^2}$.

LIST OF REFERENCES

LIST OF REFERENCES

- [1] Kwakernaak, N., and Sivan, R., Linear Optimal Control Systems, Wiley-Interscience, John Wiley and Sons, New York, 1972.
- [2] Bryson, A.E., Jr. and Ho, Y.C., Applied Optimal Control, Waltham, Mass., Ginn and Company, 1969.
- [3] Athans, M., and Falb, P.L., Optimal Control, New York: McGraw-Hill, 1966.
- [4] Skelton, R.E., "Cost Decomposition of Linear Systems with Application to Model Reduction," Int. J. Control, Vol. 32, No. 6, 1980, pp. 1031-1055.
- [5] Skelton, R.E., and Gregory, C.Z., "Measurement Feedback and Model Reduction by Modal Cost Analysis," Proceedings Joint Automatic Control Conference, Denver 1979.
- [6] Skelton, R.E., and Hughes, P.C., "Modal Cost Analysis for Linear Matrix-Second-Order Systems," Journal of Dynamic Systems, Measurement, and Control, Vol. 102, Sept. 1980.
- [7] Skelton, R.E., and Yousuff, A., "Component Cost Analysis of Large Scale Systems," Int. J. Control, Vol. 37, No. 2, 1983, pp. 285-304.
- [8] Johnson, Timothy, L., Athans, M., and Skelton, G.B., "Optimal Control-Surface Locations for Flexible Aircraft," IEEE Trans. on Auto. Control, Vol. AC-16, No. 4, August 1971.
- [9] Aidarous, S.E., "On the Structure of the Control Subsystem for Stochastic Distributed Parameter Systems," Lecture Notes in Control and Information Sciences, Distributed Parameter Systems: Modeling and Identification, Proc. of the IFIP Working Conf. Rome, Italy, June 21-24, 1976.
- [10] Aidarous, S.E., Gevers, M.R., and Installé, M.J., "Optimal pointwise discrete control and controllers' allocation strategies for stochastic distributed systems," Int. J. Control, 1975, Vol. 24, No. 4, pp. 493-508.
- [11] Ichikawa, A., and Ryan, E.P., "Sensor and Controller Location Problems for Distributed Parameter Systems," Automatica, Vol. 15, pp. 347-352, 1979.

- [12] Juang, J.N., and Rodriguez, G., "Formulations and Applications of Large Structure Actuator and Sensor Placements," Second VPI & SU/AIAA Symposium on: Dynamics and Control of Large Flexible Spacecraft, June 21-23, 1979.
- [13] Skelton, R.E., and Chiu, D., "Optimal Selection of Inputs and Outputs in Linear Stochastic Systems," to appear in Journal of Astronautical Sciences, 1983.
- [14] Chiu, D., and Skelton, R.E., "Selecting Measurements and Controls in LQG Problems," Proc. of NCKU/AAS International Symposium on Engr. Sciences and Mechanics, Tainan, Taiwan, Dec. 1981.
- [15] Chiu, D., "Optimal Sensor/Actuator Selection, Number, and Placement for Linear Stochastic Systems," Purdue University Technical Report, Contract No. 955369, West Lafayette, IN, May 1981.
- [16] Skelton, R.E., and DeLorenzo, M.L., "Selection of Noisy Actuators and Sensors in Linear Stochastic Systems," to appear in Journal of Large Scale Systems, Theory and Applications, North-Holland Publishing Company.
- [17] Yu, T.K., and Seinfeld, J.H., "Observability and Optimal Measurement Location in Linear Distributed Parameter Systems," Int. J. Control, 1973, Vol. 18, No. 4, pp. 785-799.
- [18] Chen, W., and Seinfeld, J., "Optimal Location of Process Measurements," Int. J. Control, 1975, Vol. 21, No. 6, 1003-1014.
- [19] Aidarous, S.E., Gevers, M.R., and Installé, M.J., "Optimal sensors' allocation strategies for a class of stochastic distributed systems," Int. J. Control, 1975, Vol. 22, No. 2, pp. 197-213.
- [20] Aidarous, S.E., Gevers, M.R., and Installé, M.J., "On the Asymptotic Behavior of Sensors' Allocation Algorithm in Stochastic Distributed Systems," Proc. of the IFIP Working Conf., Rome, Italy, June 21-24, 1976.
- [21] Kumar, S., and Seinfeld, J.H., "Optimal Location of Measurements for Distributed Parameter Estimation," IEEE Trans. Actions on Automatic Control, Vol. AC-23, No. 4, August 1978.
- [22] Omatu, S., Koide, S., and Soeda, T., "Optimal Sensor Location Problem for a Linear Distributed Parameter System," IEEE Trans. on Auto. Control, Vol. AC-23, No. 4, August 1978.
- [23] Wei, Y., and Wu, A., "Optimization of Large Structure Sensor Designs," Proc. 13th Asilomar Conf. Pacific Grove, Ca., Nov. 1979.
- [24] Arbel, Ami, "Sensor Placement in Optimal Filtering and Smoothing Problems," IEEE Trans. on Auto. Control, Vol. AC-27, No. 1, Feb. 1982.

- [25] Colantuoni, G., and Padmanabhan, L., "Optimal Sensor Selection in Sequential Estimation Problems," Int. J. Control, 1978, Vol. 28, No. 6, pp. 821-845.
- [26] Curtain, R.F., and Ichikawa, A., "Optimal Location of Sensors for Filtering for Distributed Systems," Proc. of the IFIP Working Conf., Rome, Italy, June 21-24, 1976.
- [27] Amouroux, M., Babary, J.P., and Malandrakis, C., "Optimal Location of Sensors for Linear Stochastic Distributed Parameter Systems," Proc. of the IFIP Working Conf., Rome, Italy, June 21-24, 1976.
- [28] Morari, Manfred and O'Dowd, M.J., "Optimal Sensor Location in the Presence of Non-Stationary Noise," Automatica, Vol. 16, pp. 463-480, 1980.
- [29] Amouroux, M., and Babary, J.P., "Optimal Pointwise Control for a Class of Distributed Parameter Systems," Proc. 6th IFAC Congress, Part I, Boston, Mass., 1975.
- [30] Kosut, R.L., Arbel, A., and Kessler, K.M., "Optimal Sensor System Design for State Reconstruction," IEEE Trans. on Auto. Control, Vol. AC-27, No. 1, Feb. 1982.
- [31] Athans, M., "On the Determination of Optimal Costly Measurement Strategies for Linear Stochastic Systems," Automatica, Vol. 8, pp. 397-412, 1972.
- [32] Herring, K.D., and Melsa, J.L., "Optimum Measurements for Estimation," IEEE Trans. on Automatic Control, Vol. AC-19, pp. 264-266, June 1974.
- [33] El-Sayed, M.L., and Krishnaprashad, "Information Measures in Decentralized Systems," Allerton Conf., 1979.
- [34] Vanbeveren, Y., and Gevers, M.R., "On Optimal and Suboptimal Actuator Selection Strategies," IEEE Trans. on Auto. Control, Vol. AC-21, pp. 382-385, June 1976.
- [35] Martin, J.E., "Dynamic Selection of Actuators for Lumped and Distributed Parameter Systems," IEEE Transactions on Auto. Control, Vol. AC-24, No. 1, February 1979.
- [36] Mehra, R.K., "Optimization of Measurement Schedules and Sensor Designs for Linear Dynamic Systems," IEEE Transactions on Automatic Control, Vol. AC-21, No. 1, February 1976.
- [37] Buhariwala, K.J., "Investigation of Sensor Placement in the Control of Large Flexible Space Structures," B.S. Thesis, Division of Engineering Science, University of Toronto, April 1980.

- [38] Ewing, D.J., Jr., and Higgins, T.J., "A Sensor Location Algorithm for Distributed-Parameter Systems," Ninth Annual Allerton Conf. on Circuit and System Theory, 1971.
- [39] Caravani, P., and Pillo, D.G., "Optimal Location of a Measurement Point in a Diffusion Process," Proc. 6th IFAC Congress, Boston, Mass., 1975, (Part I).
- [40] Arbel, Ami and Gupta, N.K., "Optimal Actuator and Sensor Locations in Oscillatory Systems," Proc. 13th Asilomar Conf., Pacific Grove, Ca., Nov. 1979.
- [41] Arbel, Ami, "Controllability Measures and Actuator Placement in Oscillatory Systems," Int. J. Control, 1981, Vol. 33, No. 3, pp. 565-574.
- [42] Vandervelde, W., and Carigan, C., "Number and Placement of Control System Actuators Considering Possible Failures," American Control Conference, Arlington, Va., June, 1982.
- [43] Lindberg, R.E., and Longman, R.W., "On the Number and Placement of Actuators for Independent Modal Space Control," to appear, Journal Guidance Dynamics and Control.
- [44] Skelton, R.E., and Delorenzo, M.L., "On Selection of Weighting Matrices in the LQG Problem," 20th Annual Allerton Conference on Circuit and System Theory. Allerton, Ill., Oct. 1982.
- [45] Skelton, R.E., and DeLorenzo, M.L., "Weight Selection for Covariance Constrained LQG Regulators: Large Space Structure Application," submitted to Automatica.
- [46] Athans, M., "The Matrix Minimum Principle," Inform. Control, Vol. 11, No. 5, pp. 592-605, 1968.
- [47] Athans, M., "The Role and Use of the Stochastic Linear-Quadratic Gaussian Problem in Control System Design," IEEE Trans. on Auto Control, Vol. AC-16, No. 6, Dec. 1971.
- [48] Makila, P.M., Westerlund, T., and Toivonen, H.T., "Constrained Linear Quadratic Gaussian Control," 21st IEEE Conference on Decision and Control, Orlando, Fl., Dec. 1982.
- [49] Toivonen, H., "Variance-Constrained Self-Tuning Control," to appear in Automatica.
- [50] Broyden, C.G., "A Class of Methods for Solving Nonlinear Simultaneous Equations," Mathematics of Computations, Vol. 19, Number 92, p. 577-593, October 1965.
- [51] Harvey, A.H., and Stein, G., "Quadratic Weights for Asymptotic Properties," IEEE Trans. on Auto Control, Vol. AC-23, No. 3, June 1978.

- [52] Grimble, M.J., and Johnson, M.A., "Asymptotic Methods in the Selection of LQP Optimal Control Weighting Matrices," IFAC Workshop on Singular Perturbations and Robustness of Control Systems, Ohrid Yugoslavia, July 1982.
- [53] Shaked, U., "The Asymptotic Behavior of the Root-Loci of Multi-variable Optimal Regulators," IEEE Trans. on Auto., Vol. AC-23, No. 3, June 1978.
- [54] Hartmann, Harvey, Mueller, "Optimal Linear Control," Technical Report No. 1, Office of Naval Research ONR CR215-238-1, Arlington, Va., December 1975.
- [55] McEwen, R.S., and Looze, D.P., "Quadratic Weight Adjustment for the Enhancement of Feedback Properties," American Control Conference, Arlington, Va., June 1982.
- [56] Safonov, M.G., "Choice of Quadratic Cost and Noise Matrices and the Feedback Properties of Multiloop LQG Regulators," Proc. of 13th Asilomar Conf. on Circuits, Sys., and Comp., Pacific Grove, Ca., Nov. 5-7, 1979.
- [57] Sesak, J.R., Likins, P., and Coradetti, T., "Flexible Spacecraft Control by Model Error Sensitivity Suppression," Journal of Astronautical Sciences, Vol. XXXVII, No. 2, pp. 131-156, April-June, 1979.
- [58] Sesak, J.R., et al., "Constrained Optimal Compensation Design for Large Flexible Spacecraft Control," Proc. 19th IEEE Conference on Decision and Control, Albuquerque, N.M., Dec. 1980.
- [59] Zoutendijk, G., Methods of Feasible Directions, Elsevier Publishing Co., Amsterdam, 1960.
- [60] Luenberger, David G., Introduction to Linear and Non-linear Programming, Addison-Wesley Publishing Company, Reading, Mass., 1973.
- [61] Hadley, G., Nonlinear and Dynamic Programming, Addison-Wesley Publishing Co., Reading, Mass., 1964.
- [62] Berkovitz, L.D., Optimal Control Theory, Springer-Verlag, New York, N.Y., 1974.
- [63] Chen, C.T., Introduction to Linear System Theory, Holt, Rinehart and Winston, Inc., 1970.
- [64] Yousuff, A., and Skelton, R.E., "Balanced Controller Reduction," IEEE Trans. Auto. Control, to appear, AC-29, Vol. 3, March 1984.
- [65] Hughes, P.C., and Skelton, R.E., "Stability, Controllability and Observability of Linear Matrix-Second-Order Systems," Proceedings JACC, June 1979, pp. 56-62.

- [66] Sokolnikoff, I.S., and Redheffer, R.M., Mathematics of Physics and Modern Engineering, McGraw-Hill, Inc., 1966.
- [67] Likins, P.W., "Dynamics and Control of Flexible Space Vehicles," Jet Propulsion Laboratory, TR32-1329, Rev. 1, Jan. 1970.
- [68] Balas, M.J., "Trend in Large Space Structure Control Theory: Fondest Hopes, Wildest Dreams, IEEE Transactions on Automatic Control, Vol. AC-27, No. 3, June 1982.
- [69] Kissell, G.J., and Lin, J.G., "Spillover Prevention via Proper Synthesis/Placement of Actuators and Sensors," American Control Conference, Arlington, Va., June 1982.
- [70] Wu, Y.W., Rice, R.B., and Juang, J.N., "Sensor and Actuator Placement for Large Flexible Space Structures," JACC, 1979, Denver, Colorado.
- [71] Johnson, T.L., "Principles of Sensor and Actuator Location in Distributed Systems," Proc. of NCKU/AAS International Symposium on Engineering Sciences and Mechanics, Tainan, Taiwan, Dec. 1981.
- [72] Gupta, N.K., and Hall, W.E., Jr., "Design and Evaluation of Sensor Systems for State and Parameter Estimation," J. Guidance and Control, Vol. 1, No. 6, Nov.-Dec. 1978.
- [73] Balas, M.J., "The Galerkin Method and Feedback Control of Linear Distributed Parameter Systems," Journal Math. Analysis and Application, Vol. 91, No. 2, Feb. 1983, pp. 527-546.
- [74] Hadley, G., Linear Algebra, Addison-Wesley, 1961.
- [75] Yousuff, A., and Skelton, R.E., "A Solar Optical Telescope Controller Design by Component Cost Analysis," Proc. Symposium IFAC/CDSS CERT/DERA, Toulouse, France, June 29-July 2, 1982.
- [76] Kalaba, R., and Spingarn, K., "Design of Linear Regulators with Energy Constraints," 19th IEEE Conf. on Decision and Control, Albuquerque, N.M., Dec. 10-12, 1980.

APPENDICES

Appendix A: Proofs

Theorem 1 proof:

For the proof of equations (5.4a,b) begin with the closed-loop system representation of (3.3) *partially* repeated here for convenience.

$$(A1) \quad \left\{ \begin{array}{l} \dot{x} = Ax + Bu ; \quad x = (x^T \hat{x}^T)^T ; \quad w = (w^T v^T)^T \\ x \in \mathbb{R}^{2n} , \quad w \in \mathbb{R}^{p+l} \\ y = Cx ; \quad y = [y^T u^T]^T ; \quad y \in \mathbb{R}^{m+k} \\ v = E_y^T Q y ; \quad Q = \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix} \\ \text{and} \\ E_w\{w(t) w^T(\tau)\} = W \delta(t-\tau) ; \quad W = \begin{bmatrix} W & 0 \\ 0 & V \end{bmatrix} \end{array} \right.$$

Then, the following holds:

$$(A2) \quad \left\{ \frac{\partial y^T Q y}{\partial y_i} = \frac{\partial y^T Q y}{\partial y} \frac{\partial y}{\partial y_i} = 2 y^T Q e_i ; \quad i = 1, \dots, k \right.$$

where $e_i \in \mathbb{R}^{m+k}$ and has a zero in all entries except for a 1 in the i^{th} entry. By the same token the following expression is also true.

$$(A3) \quad \left\{ \frac{\partial y^T Q u}{\partial u_i} = 2 y^T Q e_i ; \quad i = k+1, \dots, k+m \right.$$

Substituting A2 and A3 into the definitions of (5.3) yields:

$$(A4) \quad \begin{cases} v_i^u = E_{\infty}\{y^T Q e_i u_i\} ; & i = k+1, \dots, k+m \\ v_i^y = E_{\infty}\{y^T Q e_i y_i\} , & i = 1, \dots, k \end{cases}$$

Letting e_i^y and e_i^u be k and m dimensional versions of e_i and using the partitions of y and Q in (A1), (A4) can be rewritten as follows:

$$(A5) \quad \begin{cases} v_i^u = E_{\infty}\{u^T R e_i^u u_i\} & i = 1, \dots, m \\ v_i^y = E_{\infty}\{y^T Q e_i^y y_i\} & i = 1, \dots, k \end{cases}$$

Using $\text{tr}[AB] = \text{tr}[BA]$, $u = G\hat{x}$, $y = Cx$, gives:

$$(A6) \quad \begin{cases} v_i^u = E_{\infty}\text{tr}[e_i^u g_i^T \hat{x} \hat{x}^T G^T R] & i = 1, \dots, m \\ v_i^y = E_{\infty}\text{tr}[e_i^y c_i^T x x^T C^T Q] , & i = 1, \dots, k \end{cases}$$

where g_i^T is the i^{th} row of G and c_i^T is the i^{th} row of C . The matrices $e_i^u g_i^T$ and $e_i^y c_i^T$ can be recognized as matrices of all zero rows except for the i^{th} row of G or C respectively. Therefore, using this fact a little additional linear algebra produces:

$$(A7) \quad \begin{cases} v_i^u = E_{\infty}[G \hat{x} \hat{x}^T G^T R]_{ii} ; & i = 1, \dots, m \\ v_i^y = E_{\infty}[C x x^T C^T Q]_{ii} ; & i = 1, \dots, k \end{cases}$$

Taking the expectation of A7 and substituting in the results of (3.5) and (3.7) for the steady state variances of \hat{x} and x gives the desired result

$$(A8) \quad \begin{cases} v_i^u = [G\hat{X}G^TR]_{ii}, & i = 1, \dots, m \\ v_i^y = [C(P+\hat{X})C^TQ]_{ii}, & i = 1, \dots, k \end{cases}$$

For the proof of (5.4c-g) begin by recognizing that

$$(A9) \quad v_i^w = \begin{cases} v_i^w, & i = 1, \dots, p \\ v_i^v, & i = p+1 \dots p+l \end{cases}$$

where v_i^w is defined as follows:

$$(A10) \quad v_i^w \triangleq \lim_{t \rightarrow \infty} E \left\{ \frac{1}{2} \frac{\partial}{\partial w_i} (y^T Q y) w_i \right\};$$

Using the known solution for $\dot{x} = Ax + Bw$ the solution for y becomes

$$(A11) \quad y = Ce^{At}x(0) + C \int_0^t e^{A(t-\tau)} Bw(\tau) d\tau$$

Therefore, y can be considered a composite function of t and $w(t)$. [66]

Applying the chain rule for composite functions to the partial derivation in (A10) gives the following:

$$\frac{1}{2} \frac{\partial}{\partial w_i} (y^T Q y) = y^T Q \frac{\partial y}{\partial w_i}$$

Now, (A10) becomes:

$$(A12) \quad v_i^w = \lim_{t \rightarrow \infty} E \left\{ y^T Q \frac{\partial y}{\partial w_i} w_i \right\}$$

The term $\frac{\partial y}{\partial w_i} w_i$ can be interpreted as the contribution the i^{th} component of w is making to y . Mathematically this can be expressed as follows:

$$(A13) \quad \frac{\partial y}{\partial w_i} w_i = C \int_0^t e^{A(t-\sigma)} b_i w_i(\sigma) d\sigma$$

where b_i is the i^{th} column of B . Now, substituting (A13) and (A11) into (A12) gives:

$$(A14) \quad v_i^w = \lim_{t \rightarrow \infty} E \left\{ \left[x^T(0) e^{A^T t} C^T + \int_0^t w^T(\tau) B^T e^{A^T(t-\tau)} d\tau C^T \right] \right. \\ \left. QC \int_0^t e^{A(t-\sigma)} b_i w_i(\sigma) d\sigma \right\}$$

Upon multiplying out, v_i^w becomes

$$(A15) \quad v_i^w = \lim_{t \rightarrow \infty} E \left\{ \int_0^t x^T(0) e^{A^T t} C^T Q C e^{A(t-\sigma)} b_i w_i(\sigma) d\sigma + \right. \\ \left. \int_0^t \int_0^t w^T(\tau) B^T e^{A^T(t-\tau)} C^T Q C e^{A(t-\sigma)} b_i w_i(\sigma) d\tau d\sigma \right\}$$

Using $\text{tr}[AB] = \text{tr}[BA]$ and interchanging the integral and the expectation operators gives

$$(A16) \quad v_i^w = \lim_{t \rightarrow \infty} \text{tr} \left\{ \int_0^t b_i E\{w_i(\sigma) x^T(0)\} e^{A^T t} C^T Q C e^{A(t-\sigma)} d\sigma \right. \\ \left. + \int_0^t \int_0^t b_i E\{w_i(\sigma) w^T(\tau)\} B^T e^{A^T(t-\tau)} C^T Q C e^{A(t-\sigma)} d\tau d\sigma \right\}$$

but from the assumed noise characteristics of (1.2)

$$E(w_i(\sigma)x^T(0)) = 0$$

$$E\{w_i(\sigma)w^T(\tau)\} = w^{iT} \delta(\tau-\sigma) ; \quad \text{where } w^{iT} \text{ is the } i^{\text{th}} \text{ row of } w$$

Therefore (A16) becomes:

$$(A17) \quad v_i^w = \lim_{t \rightarrow \infty} \text{tr} \int_0^t b_i w^{iT} B^T e^{A^T(t-\sigma)} C^T Q C e^{A(t-\sigma)} d\sigma$$

Pulling out the constant terms and using the identity $\text{tr}(AB) = \text{tr}(BA)$ again:

$$(A18) \quad v_i^w = \lim_{t \rightarrow \infty} \text{tr} [B^T \int_0^t e^{A^T(t-\sigma)} C^T Q C e^{A(t-\sigma)} d\sigma b_i w^{iT}]$$

Let S be defined by:

$$(A19) \quad S = \lim_{t \rightarrow \infty} \int_0^t e^{A^T(t-\sigma)} C^T Q C e^{A(t-\sigma)} d\sigma$$

Using this definition, A18 becomes:

$$(A20) \quad \therefore v_i^w = \text{tr}[B^T S b_i w^{iT}]$$

Equation A20 can be written in the more convenient form:

$$(A21) \quad v_i^w = [B^T S B w]_{ii}$$

Since A is a stable matrix as a result of (1.3), S will approach a constant matrix as $t \rightarrow \infty$. (i.e. $\dot{S} \rightarrow 0$). Make the following change of variable. $(t-\sigma) = (\tau-t)$ in (A19):

$$(A22) \quad \lim_{t \rightarrow \infty} \int_t^{2t} e^{A^T(\tau-t)} C^T Q C e^{A(\tau-t)} d\tau = S$$

Now differentiating A22 with respect to t and making use of Leibniz's rule gives

$$(A23) \quad \dot{S} = \lim_{t \rightarrow \infty} \left[2e^{A^T t} C^T Q C e^{A t} - C^T Q C + \int_t^{2t} \frac{d e^{A^T(\tau-t)}}{dt} C^T Q C e^{A(\tau-t)} d\tau \right. \\ \left. + \int_t^{2t} e^{A^T(\tau-t)} C^T Q C \frac{d e^{A(\tau-t)}}{dt} d\tau \right]$$

Now, applying the property of state transition matrices that

$$\frac{d e^{A(\tau-t)}}{dt} = -e^{A(\tau-t)} A$$

to A23 gives

$$(A24) \quad \dot{S} = \lim_{t \rightarrow \infty} \left[2e^{A^T t} C^T Q C - C^T Q C - A^T \int_t^{2t} e^{A^T(\tau-t)} C^T Q C e^{A(\tau-t)} d\tau \right. \\ \left. - \int_t^{2t} e^{A^T(\tau-t)} C^T Q C e^{A(\tau-t)} d\tau A \right]$$

Letting $t \rightarrow \infty$ in (A24), substituting in (A22) and remembering that $\lim_{t \rightarrow \infty} e^{A^T t} \rightarrow 0$ for stable A gives

$$(A25) \quad 0 = C^T Q C + A^T S + S A$$

The steady state Lyapunov equation (A25) is a $2n$ dimensional equation which by direct substitution in (A25) can be shown to have the following solution

$$(A26) \quad S = \begin{bmatrix} K+L & -L \\ -L & L \end{bmatrix}$$

where K is the control Riccati solution of (3.2c) and L satisfies:

$$(A27) \quad L(A-FM) + (A-FM)^T L + G^T R G = 0 \quad (\text{i.e. 5.4g})$$

Therefore, using the appropriate partitions of (A21) the formulas (5.4d-f) are established. ###

Proof of Property 1:

From [1] it is known that the steady-state cost functional V of (A1) can be written in the following forms

$$(A28) \quad V = \text{tr}[XC^T Q C]$$

and

$$(A29) \quad V = \text{tr}[S B W B^T]$$

where

$$(A30) \quad A X + X A^T + B W B^T = 0$$

and

$$(A31) \quad SA + A^T S + C^T Q C = 0 \quad (\text{i.e. A25})$$

The form of the $2n$ dimensional matrix S has already been given in (A26) and by direct substitution in A30 the $2n$ dimensional matrix X can be shown to have the following form

$$(A32) \quad X = \begin{bmatrix} P + \hat{X} & \hat{X} \\ \hat{X} & \hat{X} \end{bmatrix}$$

where P is the filter Riccati solution for (3.2d) and \hat{X} is the estimate variance matrix of (3.6). Using $\text{tr}[AB] = \text{tr}[BA]$, (A28) and (A29) can be rewritten as

$$(A33) \quad \nu = \text{tr}[CXC^T Q]$$

$$(A34) \quad \nu = \text{tr}[B^T S B W]$$

Using the partitions of A and B defined in (3.3) and the partitions of S and X defined in (A26) and (A32), (A33) and (A34) can be rewritten as:

$$(A35) \quad \nu = \text{tr}[C(P + \hat{X})C^T Q] + \text{tr}[G\hat{X}G^T R]$$

$$(A36) \quad \nu = \text{tr}[D^T (K + L) D W] + \text{tr}[F^T L F V]$$

Substituting the expressions for 5.4 into 5.5 and remembering that $D = [B \ D_0]$ gives the following:

$$(A37) \quad \nu = \sum_{i=1}^m [G\hat{X}G^T R]_{ii} + \sum_{i=1}^k [C(P+\hat{X})C^T Q]_{ii}$$

$$(A38) \quad \nu = \sum_{i=1}^p [D^T(K+L)DW]_{ii} + \sum_{i=1}^l [F^T L F V]_{ii}$$

By the definition of the trace operation, (A35) is equivalent to (A37) and (A36) is equivalent (A38). Therefore Property 1 is proved. ###

Proof of Property 2:

For diagonal R, Q, W and V along with the definitions for G and F in (3.21), ν_i^u , ν_i^y , ν_i^w and ν_i^v can be written as follows:

$$(A39) \quad \nu_i^u = b_i^T K \hat{X} K b_i r_i^{-1}$$

$$(A40) \quad \nu_i^y = c_i^T (P + \hat{X}) c_i$$

$$(A41) \quad \nu_i^w = d_i^T (K + L) d_i w_i \quad (w_i = i^{\text{th}} \text{ diagonal element of } W)$$

$$(A42) \quad \nu_i^v = m_i^T P L P m_i v_i^{-1} \quad (v_i = i^{\text{th}} \text{ diagonal element of } V)^{-1}$$

($m_i^T = i^{\text{th}}$ row of M)

Under the conditions of (1.3) the matrices K, P, \hat{X} and L are known to exist and be at least positive semi-definite. Therefore, since ν_i^u , ν_i^y , ν_i^w , and ν_i^v are quadratic expressions which use these matrices or symmetric products of these matrices as weights, they can *never* be negative. (Part (a) proved)

If (A,C) is observable, K (3.2c) is known to be positive definite therefore, assuming $d_i \neq 0$ and remembering $w_i > 0$, $d_i^T K d_i w_i$ will always be positive and from (A41) so will v_i^w . (Part (b) proved).

If $(A+BG,F)$ is controllable, then \hat{X} in (3.6) is known to be positive definite, this fact coupled with the fact that K is positive definite when (A,C) is observable guarantees that $b_i^T K \hat{X} K b_i$ and $c_i^T \hat{X} c_i$ will always be positive provided $b_i, c_i \neq 0$. Therefore, as indicated by (A39) and (A40) v_i^u and v_i^y will always be positive. (Proof of part c)

If (A,D) is controllable, P (3.2d) is known to be positive definite. Therefore, $c_i^T P c_i > 0$ always and from (A40), under this condition $v_i^y > 0$. (Proof of part d)

If $(A-FM, G)$ is observable then the matrix L defined by (5.4g) is known to be positive definite. This fact coupled with the fact that P is positive definite when (A,D) is controllable means that $m_i^T P L P m_i > 0$ and $d_i^T L d_i > 0$ as long as $m_i, d_i \neq 0$. Therefore, from (A41) and (A42) $v_i^w > 0$, $v_i^v > 0$. (Proof of part e)

###

Proof of Property 3:

Under the state transformation $x = Tq$, $|T| \neq 0$, where x is defined to be the state of the system $S(n,k,m,l)$ (i.e. (1.1)-(1.3)) controlled by the LQG controller of (3.2), the following identities hold: (\sim specifies the matrices in the transformed system)

$$(A43) \quad \tilde{A} = T^{-1}AT; \quad \tilde{B} = T^{-1}B; \quad \tilde{C} = CT; \quad \tilde{M} = MT$$

$$(A44) \quad \tilde{K} = T^T K T ; \quad \tilde{P} = T^{-1} P T^{-T} ; \quad \tilde{L} = T^T L T ; \quad \tilde{\hat{X}} = T^{-1} \hat{X} T^{-T}$$

Using the definitions for G and F in (3.2) and substituting the identities of (A43) and (A44) into (5.4) the transformation matrices T , T^T are exactly cancelled by T^{-1} , T^{-T} and non-singular state transformation is seen to have no effect on the quantities v_i^u , v_i^y , v_i^w , and v_i^v .

###

Proof of Theorem 3: Theorem 3 may be proved by showing that a system of type (3.3) operating with fewer actuators is not guaranteed to have a larger total cost.

The total cost of system (3.3) can be shown to be: [1]

$$(A45) \quad V(m, \ell) = \text{tr}[P C^T Q C + K P M^T V^{-1} M P]$$

let the total cost of the system operating with a reduced set (1 or more) of inputs be:

$$(A46) \quad V(m-1, \ell) = \text{tr}[P_R C^T Q C + K_R P_R M^T V^{-1} M P_R]$$

To prove Theorem 3, a relationship between P , P_R , K , K_R must be found.

First make the following partitionings

$$(A47) \quad B \equiv D = \begin{bmatrix} D_R & D_{RT} \\ D_{TR} & D_T \end{bmatrix} ; \quad W = \begin{bmatrix} W_R & W_{RT} \\ W_{RT}^T & W_T \end{bmatrix} ; \quad R = \begin{bmatrix} R_R & R_{RT} \\ R_{RT}^T & R_T \end{bmatrix}$$

where D_R is the D matrix for the system using a reduced set of inputs. For clarity let $W_{RT} \equiv 0$ and $R_{RT} \equiv 0$. Substituting (A47) into the P and K equations of (3.2) gives:

$$(A48) \quad PA^T + AP - PM^T V^{-1} MP + [D_R W_R D_R^T + D_T W_T D_T^T] = 0$$

$$(A49) \quad KA + A^T K - K[B_R R_R^{-1} B_R^T + B_T R_T^{-1} B_T^T]K + C^T Q C = 0$$

The equations for P_R and K_R are

$$(A50) \quad P_R A^T + A P_R - P_R M^T V^{-1} M P_R + D_R W_R D_R^T = 0$$

$$(A51) \quad K_R A + A^T K_R - K_R B_R R_R^{-1} B_R^T K_R + C^T Q C = 0$$

Now subtract (A50) from (A48) and (A51) from (A49):

$$(A52) \quad (P - P_R)A^T + A(P - P_R) - PM^T V^{-1} MP + P_R M^T V^{-1} M P_R + D_T W_T D_T^T = 0$$

$$(A53) \quad (K - K_R)A + A^T(K - K_R) - K B_R R_R^{-1} B_R^T K + K_R B_R R_R^{-1} B_R^T K_R - K B_T R_T^{-1} B_T^T K = 0$$

Add and subtract $PM^T V^{-1} P_R$ to A52 and add and subtract $K_R B_R R_R^{-1} B_R^T K$ to A53:

$$(A54) \quad (P - P_R)(A^T - \underbrace{M^T V^{-1} M P_R}_{F_R^T}) + (A - \underbrace{P M^T V^{-1} M}_F)(P - P_R) + D_T W_T D_T^T = 0$$

$$(A55) \quad (K - K_R)(A - \underbrace{B_R R_R^{-1} B_R^T K}_{G_R^T}) + (A^T - \underbrace{K_R B_R R_R^{-1} B_R^T}_{G_R^T})(K - K_R) - K B_T R_T^{-1} B_T^T K = 0$$

Adding and subtracting $K_R B_T R_T^{-1} B_T^T K$ in (A-55):

$$(A56) \quad (K-K_R)(A+BG) + (A+B_R G_R)^T (K-K_R) - K_R B_T R_T^{-1} B_T^T K = 0$$

Add and subtract $(A-F_R M)(P-P_R)$ in (A54)

Add and Subtract $(A+BG)^T (K-K_R)$ in (A56)

$$(A57) \quad (P-P_R)(A-F_R M)^T + (A-F_R M)(P-P_R) + (F_R-F) M(P-P_R) + D_T W_T D_T^T = 0$$

$$(A58) \quad (K-K_R)(A+BG) + (A+BG)^T (K-K_R) + (G_R^T B_R^T - G^T B^T)(K-K_R) - K_R B_T R_T^{-1} B_T^T K = 0$$

substituting back the values for F , F_R , G , G_R into (A57) and (A58) and adding and subtracting $K_R B_T R_T^{-1} B_T^T K_R$ in (A58) gives

$$(A59) \quad (P-P_R)(A-F_R M)^T + (A-F_R M)(P-P_R) - (P-P_R) M^T V^{-1} M(P-P_R) + D_T W_T D_T^T = 0$$

$$(A60) \quad (K-K_R)(A+BG) + (A+BG)^T (K-K_R) + (K-K_R) B R^{-1} B^T (K-K_R) - K_R B_T R_T^{-1} B_T^T K_R = 0$$

multiplying (A60) by a minus sign yields:

$$(A61) \quad (K_R-K)(A+BG) + (A+BG)^T (K_R-K) - (K_R-K) B R^{-1} B^T (K_R-K) + K_R B_T R_T^{-1} B_T^T K_R = 0$$

Equation (A59) and (A60) are standard steady-state Riccati equations.

Since the reduced system is required to satisfy (1.3) and the original system was assumed to satisfy (1.3), both $(A-F_R M)$ and $(A+BG)$ will be asymptotically stable. This means that the matrices $[P-P_R]$ and $[K_R-K]$ exist and are at least positive-semi-definite. [1]

Therefore,

$$(A62) \quad (P - P_R) \geq 0 \Rightarrow P \geq P_R$$

$$(A63) \quad (K_R - K) \geq 0 \Rightarrow K_R \geq K$$

Now, going back to A45 and A46 and subtracting A46 from A45

$$(A64) \quad V - V_R = \text{tr}[(P - P_R) C^T Q C + K P M^T V^{-1} M P - K_R P_R M^T V^{-1} M P_R]$$

If the sign of (20) is positive then the reduced set of actuators is more efficient than the full set. Therefore if a feasible situation exists for which (A64) is positive, Theorem 3 is proved. (A64) can be rewritten as follows:

$$(A65) \quad V - V_R = \text{tr}[C(P - P_R)C^T Q] + \text{tr}[M(PK P - P_R K_R P_R)M^T V^{-1}]$$

From (A65) it can be seen that $PK P \geq P_R K_R P_R$ is sufficient for $V - V_R \geq 0$. From (A62) and (A63) it can be seen that the definiteness of $PK P - P_R K_R P_R$ depends on the relative definiteness of $(P - P_R)$ and $(K_R - K)$. Therefore, $PK P - P_R K_R P_R$ could feasibly be positive definite, negative definite, indefinite, etc. Consequently, Theorem 3 is proved.

###

Proof of Theorem 3:

From [1] it is known that the LQG cost functional V (3.11) can be expressed as follows: (using the notation of (3.2))

$$(A66) \quad V = \text{tr}[K D W D^T + P G^T R G]$$

Now let V_+ equal the cost functional for the system operating with one additional sensor. Therefore,

$$(A67) \quad V_+ = \text{tr}[KDWD^T + P_+G^TRG]$$

where

$$(A68) \quad P_+A^T + AP_+ - P_+M_+^TV_+^{-1}M_+P_+ + DWD^T = 0$$

$$(A69) \quad M_+ = \begin{bmatrix} M \\ m^T \end{bmatrix}; \quad m \in R^n$$

(i.e. added column of M^T matrix)

$$(A70) \quad V_+ = \begin{bmatrix} V & 0 \\ 0 & v_+ \end{bmatrix}; \quad v_+ \in R^{1 \times 1}$$

(i.e. variance of new sensor noise)

Subtracting (A67) from (A66) gives the following:

$$(A71) \quad \Delta V \triangleq V - V_+ = \text{tr}[(P-P_+)G^TRG]$$

Equation (A71) can be rewritten as follows:

$$(A72) \quad \Delta V = \text{tr}[(P-P_+)G^TRG] = \text{tr}[\sqrt{RG}(P-P_+)\sqrt{RG}^T]$$

Therefore, if $(P-P_+)$ is at least positive semi-definite the theorem is proved.

Recall that the matrix P used here and in (3.2) is defined by the following:

$$(A73) \quad PA^T + AP - PM^T V^{-1} MP + DWD^T = 0$$

Now, subtracting (A68) from (A73) gives:

$$(A74) \quad (P-P_+)A^T + A(P-P_+) - PM^T V^{-1} MP + P_+ M_+^T V_+^{-1} M_+ P_+ = 0$$

adding $\pm PM_+^T V_+^{-1} M_+ P_+$ to (A74) yields:

$$(A75) \quad (P-P_+)(A^T - M_+^T V_+^{-1} M_+ P_+) + A(P-P_+) - PM^T V^{-1} MP + PM_+^T V_+^{-1} M_+ P_+ = 0$$

adding $\pm P_+ M_+^T V_+^{-1} M_+ P_+$, $\pm P_+ M_+^T V_+^{-1} M_+ P$ to (A75) results in:

$$(A76) \quad (P-P_+)(A^T - M_+^T V_+^{-1} M_+ P_+) + (A - P_+ M_+^T V_+^{-1} M_+)(P-P_+) - PM^T V^{-1} MP + \\ PM_+^T V_+^{-1} M_+ P_+ + P_+ M_+^T V_+^{-1} M_+(P-P_+) = 0$$

adding $\pm Pmv_+^{-1} m^T P$ to (A76), making use of (A69) and (A70) and the definition $F_+ = P_+ M_+^T V_+^{-1}$ gives:

$$(A77) \quad (P-P_+)(A - F_+ M_+)^T + (A - F_+ M_+)(P-P_+) - PM_+^T V_+^{-1} M_+(P-P_+) + \\ P_+ M_+^T V_+^{-1} M_+(P-P_+) + Pmv_+^{-1} m^T P = 0$$

collecting terms gives:

$$(A78) \quad (P-P_+)(A - F_+ M_+)^T + (A - F_+ M_+)(P-P_+) - (P-P_+)M_+^T V_+^{-1} M_+(P-P_+) + \\ Pmv_+^{-1} m^T P = 0$$

Equation (A78) is a standard matrix Riccati equation. It is well known that the solution to (A78) (i.e. $P-P_+$) is at least positive semi-definite if the matrix $(A-F_+M_+)$ is stable. The matrix $(A-F_+M_+)$ will be stable if the matrix pair (A,D) is stabilizable and the pair (A,M^+) is detectable. From the conditions (1.3) (A,B) is stabilizable and (A,M) is detectable. Therefore (A,M_+) must be detectable since adding a row to M (i.e. generating M_+ cannot effect the detectability of (A,M) .

###

Appendix B: ν_i^V and the Chen-Seinfeld Switching-Function

In [1] it was shown that the switching function for the extended Chen-Seinfeld method of optimal selection of sensors in systems of type (1.1) was

$$(B1) \quad \sigma_{c_i}^S = \text{tr}(\hat{P}m_i V^{-1} m_i^T \hat{P} \Lambda_2)$$

where m_i is a column of M^T and Λ_2 is defined by:

$$(B2) \quad \Lambda_2(A - \hat{P}M^T V^{-1}M) + (A^T - M^T V^{-1}M\hat{P})\Lambda_2 + KBR^{-1}B^T K = 0$$

where \hat{P} is defined by:

$$(B3) \quad \hat{P}A^T + \hat{A}\hat{P} - \sum_{i=1}^{\ell} q_i \hat{P}m_i V^{-1} m_i^T \hat{P} + BWB^T = 0$$

and

$$q_i = \begin{cases} 1 & \text{if } i^{\text{th}} \text{ sensor is to be used} \\ 0 & \text{if } i^{\text{th}} \text{ sensor is to be deleted} \end{cases}$$

The expression for ν_i^V is given by (5.4f), using $F = PM^T V^{-1}$ gives,

$$(B4) \quad \nu_i^V = [V^{-1}MPLPM^T]_{ii} = v_{ii}^{-1} m_i^T PLPm_i \quad (\text{assuming diagonal } V)$$

where

$$(B5) \quad L(A - PM^T V^{-1}M) + (A^T - M^T V^{-1}M^T P)L + KBR^{-1}B^T K = 0$$

$$(B6) \quad PA^T + AP - PM^T V^{-1}MP + BWB^T = 0$$

using the $\text{tr } AB = \text{tr } BA$ (B1) becomes:

$$(B7) \quad \sigma_{c_i}^S = \text{tr}[V^{-1}m_i^T \hat{P} \Lambda_2 \hat{P}m_i] = m_i^T \hat{P} \Lambda_2 \hat{P}m_i \text{tr} V^{-1}$$

comparing eqs. (B2), (B3) and (B7) to eqs. (B4), (B5) and (B6) it is

apparent that, with the exception of the trace operation on V^{-1} in (B7), V_i^V is equivalent to calculating $\sigma_{c_i}^S$ for the system with *all* admissible measurements present. (i.e. all $q_i = 1 \forall i$ in (B3)).

Appendix C: Conservative Step Size Argument

In this appendix an argument is presented for the conservative nature of $\left[\frac{E_{\infty} u_i^2}{u_i^2} \right]^{1/2}$ in the update equations for LQGWS. The identification of a conservative step size is fundamental to the success of LQGWS.

Substituting (6.22) with $PWR = 1/2$ into equation (3.19) gives the following result:

$$(C1) \quad r_{ii}(j+1) = \frac{\sqrt{b_i^T K(j) \hat{X}(j) K(j) b_i}}{u_i}$$

where the argument (j) after the matrices K and \hat{X} implies that the values of Q and R at the j^{th} iteration were used to determine the matrices, adopting this notation, the steady-state means square value of the i^{th} control at the $j^{th}+1$ iteration may be expressed as follows

$$(C2) \quad E_{\infty}\{u_i^2\} \Big|_{j+1} = r_{ii}^{-2}(j+1) b_i^T K(j+1) \hat{X}(j+1) K(j+1) b_i$$

Substituting (C-1) into (C-2) yields:

$$(C3) \quad E_{\infty}\{u_i^2\} \Big|_{j+1} = u_i^2 \frac{b_i^T K(j+1) \hat{X}(j+1) K(j+1) b_i}{b_i^T K(j) \hat{X}(j) K(j) b_i}$$

Therefore, if

$$(C4) \quad E_{\infty}\{u_i^2\} \Big|_{j+1} > u_i^2 \text{ and } b_i^T K(j+1) \hat{X}(j+1) K(j+1) b_i > b_i^T K(j) \hat{X}(j) K(j) b_i$$

then, from (C-3), (C-1) is a conservative step. Conversely, if

$$(C5) \quad E_{\infty}\{u_i^2\} \Big|_j < u_i^2 \text{ and } b_i^T K(j+1) \hat{X}(j+1) K(j+1) b_i < b_i^T K(j) \hat{X}(j) K(j) b_i$$

then again from (C-3), (C-1) represents a conservative step size. The arguments presented in the appendix are directed toward showing that the conditions of (C-4 and (C-5) should hold when the update equation of (3.5a) is applied with $pwr = .5$. Now, assume the following situation exists

$$(C6a) \quad E_{\infty}\{u_i^2\} \Big|_j > u_i^2 \quad \forall i = 1, \dots, m$$

$$(C6b) \quad \text{System (1.1) is observable and stabilizable}$$

Applying (3.5a) with $pwr = .5$ in this situation yields:

$$(C7) \quad R(j+1) > R(j) \quad (\text{i.e. } x^T R(j+1)x > x^T R(j)x \quad \forall x: ||x|| \neq (0))$$

Under condition (C-7) it can be shown that

$$(C8) \quad K(j+1) \geq K(j) \quad (\text{i.e. } x^T K(j+1)x \geq x^T K(j)x \quad \forall x).$$

The proof is not included here, but it involves the differentiating of the control Riccati equation of the j^{th} and the $j^{\text{th}}+1$ iteration and then adding and subtracting terms until a Riccati type equation results for the difference matrix $(K(j+1) - K(j))$.

Given (C-7) and (C-8), the following conjecture is made:

$$(C9) \quad \text{conjecture: } \hat{X}(j+1) \geq \hat{X}(j)$$

Justification:

Since the system of (2.1) is assumed observable and stabilizable, \hat{X} is known to be equivalent to the following integral equation:

$$(C10) \quad \hat{X}(j) = \int_0^{\infty} e^{[A+BG(j)]t} F V F^T e^{[A+BG(j)]^T t} dt$$

It is also known that increasing R (i.e. the situation in (C-7)) results in the closed-loop system poles moving toward their open-loop positions (i.e. $G \rightarrow 0$). If the open-loop system has unstable poles (the usual case for large space structures) $R(j+1) > R(j)$ implies that some eigenvalues of $[A+BG(j+1)]$ will lie closer to the imaginary axis than any of the eigenvalues of $[A+BG(j)]$. That is $(A+BG(j+1))$ will be less stable than $[A+BG(j)]$. Therefore, in determining $\hat{X}(j+1)$ from C-10, certain terms will go to zero more slowly than they did in the $\hat{X}(j)$ calculations and therefore the infinite integral of these terms must be *larger* than the previous iteration. The conjecture of (C-9) is based upon this fact.

Continuing, if $K(j+1) > K(j)$, and $\hat{X}(j+1) > \hat{X}(j)$ it is straightforward to show the following:

$$(C11) \quad K(j+1)\hat{X}(j+1)K(j+1) \geq K(j)\hat{X}(j)K(j)$$

Given (C.6a), C.11 immediately implies the conditions of C.4 hold and therefore $PWR = .5$ in (6.22) would be a conservative step size, assuming no Q adjustments. If,

$$(C12a) \quad E_{\infty}(u_i^2) \Big|_j < u_i^2 \quad \forall i = 1, \dots, m, \text{ and}$$

$$(C12b) \quad \text{System (1.1) observable and stabilizable}$$

Using 6.22 as the update equation for the $j^{th}+1$ iteration results in

$$(C13) \quad R(j+1) < R(j)$$

and it can be shown that

$$(C14) \quad K(j+1) \leq K(j)$$

An argument that parallels the one offered for the conjecture of (C9) can be used to justify the conjecture of C15

$$(C15) \quad \text{conjecture: } \hat{X}(j+1) \leq \hat{X}(j)$$

It is then again straightforward to show that

$$(C16) \quad K(j+1) \hat{X} K(j+1) \leq K(j) \hat{X}(j) K(j)$$

Given C12a, C16 implies the conditions of C.5 hold and therefore $pwr = .5$ in (3.4a) would again be a conservative step size if no Q adjustments were made.

The arguments above present a case for the conservative nature of 6.22, $PWR = .5$ when all inputs are above or below specification and no Q adjustments (i.e. (6.29a) not used) are made. No statement is made about the situation when inputs are both above and below specification or when Q adjustments occur simultaneously or by themselves. In fact, an analysis such as described above cannot be applied to the output mean square equation (3.16) and the Q update equation because (3.16) is not an explicit function of Q . However, the experience to date has shown that $pwr = .5$ in all update situations generates a conservative step size. In fact, when *just* Q adjustments are required $pwr = 1$, has also generated a conservative step size and $pwr = .5$ automatically generates a more conservative step size than $pwr = .1$.

Appendix D: Necessary and Sufficient Conditions for $E_{\infty} u_i^2 = 0 \quad \forall Q>, R>0$.

Given the system (1.1) and the LQG controller (3.2)-(3.6), the purpose of this Appendix is to identify the necessary and sufficient conditions for

$$(D1) \quad b_i^T \hat{K} X K b_i = 0 \quad \forall Q > 0, R > 0; \text{ where } b_i = i^{\text{th}} \text{ column of } B, \\ b_i \neq 0$$

For necessity, consider that (D-1) implies that $K \geq 0$ and/or $\hat{X} \geq 0$. Furthermore,

$$(D2) \quad \begin{aligned} K \geq 0 &\Rightarrow (A, C) \text{ unobservable} \\ \hat{X} \geq 0 &\Rightarrow (A+BG, F) \text{ uncontrollable} \end{aligned}$$

Thus, the necessary condition for (D1) to hold is: (A, C) unobservable and/or (A+BG, F) uncontrollable.

For sufficiency, assume first that (A, C) is unobservable and (1.1) has been placed in observable canonical form. (i.e. $x = Tx$, $|T| \neq 0$) Then, the following is true,

$$(D3) \quad \dot{x} = \begin{bmatrix} A_{11} & 0 \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u + \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} w;$$

$$(D4) \quad y = [C_1 \quad 0] \text{ and } z = Mx + v$$

$$(D5) \quad K = \begin{bmatrix} \bar{K}_{11} & 0 \\ 0 & 0 \end{bmatrix} \quad (\text{control Riccati Solution; } K_{11} > 0)$$

where,

$$(D6) \quad A = T^{-1}AT; \quad B = T^{-1}B = [b_1, \dots, b_m]; \\ D = T^{-1}D; \quad [C_1 \quad 0] = CT; \quad M = MT; \quad K = T^T K T$$

Now, from (D3) - (D6), the following can be deduced:

$$(D7) \quad Kb_i = 0 \iff K_{11}b_{1_i} = 0$$

and since $K_{11} > 0$,

$$(D8) \quad K_{11}b_{1_i} = 0 \iff b_{1_i} = 0 \quad Q > 0, \quad R > 0$$

Conditions (D7) and (D8) lead to the following sufficient condition:

Sufficient Condition ①: If (A, C) is unobservable and the i^{th} column of B_1 is zero, then (D1) holds.

Sufficient Condition ① describes the situation when the i^{th} input has no effect on observable states of the system. To check Sufficient Condition ① directly, it would be necessary to place (1.1) in observable canonical form. This numerical burden is certainly unnecessary for LQGWS since $b_i^T K = 0$ does not depend upon the particular $Q > 0, R > 0$, the test for ① (i.e. $b_i^T K = 0$) only needs to be conducted on the first iteration of LQGWS or whenever an output weight is zeroed.

Placing the estimation dynamics

$$(D9) \quad \dot{\hat{x}} = (A + BG)\hat{x} + F(z - M\hat{x})$$

$$(D10) \quad u = G\hat{x}$$

in control canonical form (i.e. $\hat{x} = S\hat{x}$, $|S| \neq 0$ yields:

$$(D11) \quad \dot{\hat{x}} = \begin{bmatrix} L_{11} & L_{12} \\ 0 & L_{22} \end{bmatrix} \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix} + \begin{bmatrix} \overline{F}_1 \\ 0 \end{bmatrix} (z - M\hat{x})$$

$$(D12) \quad u = [G_1, G_2] \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix}; \quad \hat{x} = \begin{bmatrix} \hat{x}_{11} & 0 \\ 0 & 0 \end{bmatrix} \quad \hat{x}_{11} > 0$$

where

$$(D13) \quad \begin{bmatrix} L_{11} & L_{12} \\ 0 & L_{22} \end{bmatrix} = S^{-1}(A+BG)S; \quad \begin{bmatrix} F_1 \\ 0 \end{bmatrix} = S^{-1}F$$

$$(D14) \quad [G_1, G_2] = GS; \quad \hat{x} = S^{-1}\hat{X}S^{-T}$$

As a result of (D-11)-D-14) the following holds:

$$(D15) \quad E_{\infty} u_i^2 = r_i^{-2} b_i^T K \hat{X} K b_i = [G_1 \hat{x}_{11} G_1^T]_{ii} = g_{1i}^T \hat{x}_{11} g_{1i}$$

Equation (D15) leads directly to Condition ②.

Condition ② : If $(A+BG, F)$ is uncontrollable and g_{1i} is zero, then $b_i^T K \hat{X} K b_i = 0$.

Note that Condition ② does not imply that (D1) holds since we have not shown that $g_{1i} = 0$ for every $Q > 0, R > 0$. Indeed this condition cannot be shown unless the controllability of $(A+BG, F)$ is independent of the values of the matrices $Q > 0, R > 0$ when the rank of F is less than n . The physical significance of condition ② is that $E_{\infty} u_i^2 = 0$ LQG Controls which do not contain estimators states that are disturbed from z .

Conditions ① and ② together describe all the possible conditions for which (D1) could hold. As a result, LQGWTS performs the following sequence of steps to check (D1).

1. On the first iteration and any time a q_i is zeroed, delete any actuators associated with a zero row of $B^T K$.

2. Whenever r_i adjustments force $r_i < \epsilon/u_i^2$ delete the i^{th} actuator.

(Condition (2) then sufficient.)

The justification for step (2) results from the fact that update equation (6.22) will decrease r_i as long as $E_{\infty} u_i^2 < \mu_i^2$.

Finally, it should be noted that when an actuator is deleted it is also necessary to recalculate P (3.2e) since $D \stackrel{\Delta}{=} [B \quad D_0]$

Appendix E: Program Listings

Program LQGWTS

```

SUBROUTINE LQWTS(NRA,NA,AA,NRB,BB,NRD,NM,DD,NRW,WW,NRU,UUI,
~NRC,CC,NRM,MM,SIGMA,MU,TITLY,TITLU,TITLZ,NY,NU,NZ,FLAGS,EY,EU,
~EYNORM,ELNORM,NRP,PP,NRK,KK,NRX,XXH,EIGAFH,EIGABG,NRQ,QQ,NRR,RRI,
~WK,EPS,ITNUM)
C *****
C *** GIVEN THE FOLLOWING TIME-INVARIANT, LINEAR, STOCHASTIC SYSTEM:
C ***
C ***   DX/DT = AA*X + BB*U + DD*W
C ***   Y = CC*X
C ***   Z = MM*X + U
C ***   DD = [BB, DD ] ; WHERE DD CAN BE 0
C ***           0           0
C ***
C *** WITH THE NOISE CHARACTERISTICS:
C ***           T
C ***   E[W] = E[U] = 0 ; E[W (T)W($)] = WW * DELTA(T-$)
C ***           T
C ***   E[U (T)U($)] = UU * DELTA(T-$)
C ***
C ***   WW > 0 AND UU > 0
C ***
C *** DETERMINE THE STEADY STATE LQG CONTROLLER :
C ***           ↑
C ***   U = GG*X
C ***
C *** WHERE,
C ***           ↑           ↑           ↑
C ***   DX/DT = AA*X + BB*U + FF*(Z - MM*X)
C ***           -1           T
C ***   GG = -RR * BB * KK
C ***           T           -1
C ***   FF = PP * MM * UU
C ***           T           -1           T           T
C ***   KK*AA + AA *KK - KK*BB*RR *BB *KK + CC *QQ*CC = 0
C ***           T           T           -1           T
C ***   PP*AA + AA*PP - PP*MM *UU *MM*PP + DD*WW*DD = 0
C ***
C *** SUCH THAT ,
C ***           2
C ***   LIM E[Y ] .LE. SIGMA(I) ;      FOR ALL I=1,...,NY
C ***           I
C ***           2
C ***   LIM E[U ] .LE. MU(I) ;      FOR ALL I=1,...,NU
C ***           I
C ***
C *** ALSO, DETERMINE THE SENSOR AND ACTUATOR EFFECTIVENESS VALUES
C ***
C ***
C *** IF NO SOLUTION EXISTS , DETERMINE THE MINIMUM OUTPUT
C *** SPECIFICATIONS AND THE LQG CONTROLLER WHICH KEEPS THE STEADY-
C *** STATE MEAN-SQUARE INPUT VALUES WITHIN THEIR ORIGINAL SPECIFICA-
C *** TION (I.E. AN INPUT CONSTRAINED SOLUTION) ;
C *** OR , CONVERSELY , DETERMINE THE MINIMUM INPUT SPECIFICATIONS
C *** AND THE LQG CONTROLLER WHICH KEEPS THE STEADY-STATE MEAN-SQUARE
C *** OUTPUT VALUES WITHIN THEIR ORIGINAL SPECIFICATIONS.
C *** (I.E. AN OUTPUT CONSTRAINED SOLUTION)

```

```

C ***
C *** INPUT ARGUMENTS -
C *** AA(NA,NA),BB(NA,NU),DD(NA,NU),UU(NU,NU),VUI(NZ,NZ)(IN-
C *** VERSE OF UU),CC(NY,NA),MM(NZ,NA); SYSTEM MODEL MATRICES
C *** WITH ROW DIMENSIONS NRA,NRB,NRD,NRW,NRU,NRC,NRM RESPEC-
C *** TIVELY.
C *** NA,NU,NY,NZ,NU: VECTOR LENGTHS OF X,U,Y,Z,W RESPECTIVELY
C *** SIGMA(NY): VECTOR OF OUTPUT SPECIFICATIONS
C *** MU(NU): VECTOR OF INPUT SPECIFICATIONS
C *** TITLY(NY): VECTOR OF 10-CHARACTER OUTPUT LABELS
C *** TITLU(NU): VECTOR OF 10-CHARACTER INPUT LABELS
C *** TITLZ(NZ): VECTOR OF 10-CHARACTER SENSOR LABELS
C *** FLAGS(11): LOGICAL PROGRAM CONTROLS
C *** FLAGS(1)=.T. PRINTS PARTIAL ECHO CHECK OF INPUT DATA
C *** FLAGS(2)=.T. INPUT CONSTRAINED SEARCH USED
C *** FLAGS(3)=.F. OUTPUT CONSTRAINED SEARCH USED
C *** FLAGS(4)=.T. CHECK FILTER RICCATI SOLN
C *** FLAGS(5)=.T. READ INITIAL GUESS FOR Q-R FROM TAPE
C *** FLAGS(6)=.T. CHECK CONTROL RICCATI ^ EST. LYAP SOLN
C *** FLAGS(7)=.T. PRINT INTERMEDIATE RESULTS
C *** FLAGS(8)=.T. STORE INTERMEDIATE RESULTS
C *** FLAGS(9)=.T. PRINT FINAL RESULTS
C *** FLAGS(10) USED BY SUBROUTINE
C *** FLAGS(11) USED BY SUBROUTINE
C *** WK: REAL WORK VECTOR OF LENGTH .GE.
C *** [4*NY+3*NU+4*NA+NA+NA+NU+MAX(NA*NA,NY*NY,NU*NU)+
C *** MAX(6*NA+4*NA,NU*NU+2*NZ*NZ)]
C *** EPS: CONVERGENCE BOUND , I.E.
C ***  $\frac{2}{(1-EPS).LE.(E|U|/MU(I)).LE.(1+EPS)}$  IMPLIES CONVERGENCE
C *** I
C *** SUGGESTED RANGE .1 > EPS > .001
C *** ITNUM: MAX NUMBER OF ITERATIONS ALLOWED
C ***
C *** OUTPUT ARGUMENTS-
C ***
C *** EY(NY,1): VECTOR OF CURRENT OUTPUT MS VALUES
C *** EY(NY,2): VECTOR INDICATOR OF OUTPUT WTS. BEING UPDATED
C *** DURING AN INPUT CONST. SEARCH. (IE. EY(I,2)=1.
C *** IMPLIES UPDATE.)
C *** EU(NU,1): VECTOR OF CURRENT INPUT MS VALUES
C *** EU(NU,2): VECTOR INDICATOR OF INPUT WTS BEING UPDATED
C *** DURING AN OUTPUT CONST. SEARCH.
C *** EYNORM(NY,2): MATRIX OF THE LAST TWO NORMED OUTPUT MS VALS
C *** EUNORM(NU,2): MATRIX OF THE LAST TWO NORMED INPUT MS VALS
C *** PP(NA,NA): FILTER RICCATI SOLN, ROW DIMENSION NRP
C *** KK(NA,NA): CONTROL RICCATI SOLN, ROW DIMENSION NRK
C *** EIGABG(4*NA): REAL WORK VECTOR, EIGABG(1) CONTAINS THE EI-
C *** GEN VALUES OF AA+BB*GG
C *** EIGAFM(4*NA): REAL WORK VECTOR, EIGAFM(1) CONTAINS THE EI-
C *** GEN VALUES OF AA-FF*MM
C *** XXH(NA,NA): EST VARIANCE MATRIX, ROW DIMENSION NRX
C ***
C *** NOTES- (1) IF FLAGS(4)=.F., THE PROGRAM ASSIGNS A SET OF INITIAL
C *** WEIGHTS WHICH GIVES A COMPROMISE SOLUTION. (I.E. THE
C *** AVERAGE SPECIFICATION DEVIATION OF ALL COMPONENTS IS
C *** MINIMIZED.
C ***
C *** (2) IF BB = DD THE PROGRAM EXPECTS THE DD ARGUMENT TO BE

```

```

C ***      BB.(I.E. THE DD MATRIX IS NOT NEEDED) . THE ALGORITHM
C ***      DELETES ACTUATORS WHICH HAVE NO EFFECT ON THE OUTPUT.
C ***
C ***      (3) THE PROGRAM WRITES AND READS OCCUR ON TAPES
C ***
C ***      (4) SUBROUTINES MDROP,MAREAD,MWRITE,AND LYCHK MUST BE
C ***      DECLARED EXTERNALS. THEY ARE LOCATED AT THE BACK OF
C ***      LOGITS
C ***
C ***      (5) IF THE ALGORITHM HAS NOT CONVERGED IN THE ALLOWED
C ***      NUMBER OF ITERATIONS, IT MAY BE RESTARTED BY READ-
C ***      ING IN THE LAST VALUES OF THE APPROPRIATE UPDATE VEC-
C ***      TOR AND WEIGHTING MATRICES. ALSO, FLAGS(10) MUST BE
C ***      SET TO .T. IF BOTH Q AND R ADJUSTMENTS ARE OCCURING,
C ***      OTHERWISE FLAGS(10) MUST BE SET TO .F. .
C ***
C ***      SUBROUTINES USED: MP31,MP32,POTTER,MP3,MAREAD,MULRTR,MULRRT,
C ***      MDROP,MULT,MADD,GEIGEN,LYAP2,LYCHK,MPRINT,
C ***      MWRITE
C *****
C ***      01/20/83 MLD

```

```

      REAL AA(NRA,1),BB(NRB,1),CC(NRC,1),DD(NRD,1),WW(NRW,1),UU(NRU,1)
      ^,MM(NRM,1),SIGMA(NY),MU(NU),TITLU(NU),EY(NY,2),EU(NU,2)
      ^,EYNORM(NY,2),EUNORM(NU,2),PP(NRP,1),KK(NRK,1),QQ(NRQ,1),TITLZ(NZ
      ^),RRI(NRR,1),WK(11),EIGAFM(NA),EIGABG(NA),XXH(NRX,1)

```

```

      LOGICAL FLAGS(11),SOLN,NSOLN,SSOLN,INTER,CONST,XHAT,QZFLG,RZFLG

```

```

      INTEGER CNUM(1),RNUM(1)

```

```

1  FORMAT(1H1)
2  FORMAT(5X,/)
3  FORMAT(5X,*****):
10 FORMAT(27X,*****),/,27X,## INPUT DATA #,
   ^ECHO CHECK ##,/,27X,*****),///,5X,##NUMBER#,
   ^ OF INPUTS #,I3,/,5X,##NUMBER OF OUTPUTS #,I3,/,5X,##VALUE OF #
   ^,##EPSILON #,E11.5,/)
11 FORMAT(9X,##SENSOR#,12X,##SENSOR#,11X,##ACTUATOR#,11X,##NOISE#,
   ^11X,##ACTUATOR#,10X,##ACTUATOR#,/,5X,##NUMBER#,3X,##LABEL#,5X,
   ^EFFECTIVENESS#,4X,##NUMBER#,3X,##LABEL#,5X,##CONTRIBUTION#,4X,
   ^NUMBER#,3X,##LABEL#,5X,##EFFECTIVENESS#)
12 FORMAT(6X,F3.0,3X,A10,2X,E11.5,7X,F3.0,3X,A10,2X,E11.5,6X,F3.0,
   ^3X,A10,2X,E11.5)
13 FORMAT(6X,F3.0,3X,A10,2X,E11.5)
14 FORMAT(42X,F3.0,3X,A10,2X,E11.5,6X,F3.0,3X,A10,2X,E11.5)
15 FORMAT(/,34X,*****),/,34X,
   ^## ACTUATOR / SENSOR EFFECTIVENESS DATA ##,/,34X,*****),
   ^*****),/,/)
20 FORMAT(5X,##FLAG NUMBER:##,3X,##2#,6X,##3#,6X,##4#,6X,##5#,6X,##6#,6X
   ^,##7#,6X,##8#,6X,##9#,/,11X,##VALUE:##,1X,A5,7(2X,A5)
   ^,/)
30 FORMAT(9X,##OUTPUTS#,12X,##MEAN SQUARE#,2X,##,6X,##INPUTS#,8X,##
   ^MEAN SQUARE#,/,5X,##NUMBER#,4X,##LABEL#,7X,##SPECIFICATION#,1X,##,
   ^1X,##NUMBER#,4X,##LABEL#,7X,##SPECIFICATION#,/,41X,##)
40 FORMAT(5X,I3,4X,A10,5X,E11.5,3X,##,2X,I3,4X,A10,5X,E11.5)
41 FORMAT(5X,I3,4X,A10,5X,E11.5,3X,##)
42 FORMAT(41X,##,2X,I3,4X,A10,5X,E11.5)
50 FORMAT(5X,## THE FOLLOWING IS THE #,A6,## MATRIX.##,9X,##,/,5X,
   ^## ALL ELEMENTS SHOULD BE CLOSE TO ZERO .##,6X,##)

```

```

51 FORMAT(SX, ** THE FOLLOWING ARE THE EIGENVALUES OF #, A4, # . ** )
52 FORMAT(SX, ** THIS SPECTRUM IS INDEPENDENT OF Q AND R #, SX, ** )
53 FORMAT(48X, #NATURAL#, /, 47X, #FREQUENCY#, 7X, #DAMPING#, /, 4X, #EIGEN#
  ^, #VALUE#, 6X, #REAL#, 9X, #IMAGINARY#, 5X, # (RAD/SEC) #, 7X, #RATIO#, /)
54 FORMAT(8X, I3, 5X, E12.6, 3X, E12.6, 3X, E12.6, 3X, E12.6)
60 FORMAT(SX, ** THE FIXED COMPONENT OF THE MEAN SQUARE #, /, SX,
  ^, ** VALUE OF OUTPUT #, I3, # LABELED #, A10, 6X, ** #, /, 5X, ** EXCEEDS #
  ^, #ITS SPECIFICATION #, 17X, ** )
61 FORMAT(SX, ** NO CHOICE FOR Q AND R EXISTS . S#,
  ^#HOWN#, 7X, ** #, /, 5X, ** BELOW ARE THE FIXED COMPONENTS AND #,
  ^#THE SPEC#, 1X, ** )
62 FORMAT(/, 9X, #OUTPUTS#, 15X, #FIXED#, 12X, #MEAN SQUARE#, /, 5X,
  ^#NUMBER#, 4X, #LABEL#, 9X, #COMPONENT#, 9X, #SPECIFICATION#, /)
63 FORMAT(6X, I3, 4X, A10, 4X, E11.5, 10X, E11.5)
70 FORMAT(SX, ** INPUT #, I3, # LABELED #, A10, # HAS NO EFFECT ** ,
  ^, 5X, ** ON THE OUTPUT . IT WILL BE DELETED #, 8X, ** )
71 FORMAT(SX, ** SHOWN BELOW ARE THE INPUTS WITH THEIR#, 7X, ** #, /, 5X,
  ^# RESPECTIVE ROW NORMS IN THE GAIN MATRIX#, 5X, ** #, /, 5X, ** (I.E.
  ^ THE ROW NORMS OF BTK)#, 15X, ** )
72 FORMAT(6X, I3, 4X, A10, 5X, E11.5)
73 FORMAT(10X, #INPUTS#, 12X, #GAIN MATRIX#, /, 5X, #NUMBER#, 4X, #LABEL#, 9X
  ^, #ROW NORM#)
74 FORMAT(SX, ** INITIAL Q AND R RESULTS #, 19X, ** )
75 FORMAT(SX, ** THE ABOVE DELETED INPUTS CONSIST OF STATE#, 3X, **
  ^, /, 5X, ** ESTIMATES THAT ARE NOT DISTURBABLE FROM THE **,
  ^, 5X, ** MEASUREMENT VECTOR Z #, 22X, ** )
80 FORMAT(SX, ** ITERATION NUMBER #, I3, 24X, ** )
81 FORMAT(SX, ** #, A7, # ADJUSTMENT , EXPONENT IS #, F8.4, 3X, ** )
82 FORMAT(SX, ** SOLUTION OBTAINED#, 27X, ** )
83 FORMAT(SX, ** NO SOLUTION FOR THESE SPECIFICATIONS#, 8X, ** )
84 FORMAT(SX, ** SOLUTION NOT YET OBTAINED#, 19X, ** )
87 FORMAT(SX, ** IF THE #, A6, # SPECIFICATIONS ARE CHANGED TO ** :
  ^, /, 5X, ** THE #, A6, # MEAN SQUARE VALUES OF THIS#, 7X, ** #, /, 5X
  ^, ** ITERATION, A SOLUTION HAS BEEN FOUND . (I.E. ** #, /, 5X
  ^, ** AN #, A6, # CONSTRAINED SOLUTION .)#, 11X, ** )
88 FORMAT(SX, ** IF THE #, A6, # SPECIFICATIONS ARE CHANGED TO ** :
  ^, /, 5X, ** THE #, A6, # MEAN SQUARE VALUES OF THIS#, 7X, ** #, /, 5X
  ^, ** ITERATION, A SOLUTION HAS BEEN FOUND #, 6X, ** )
89 FORMAT(SX, ** IF THE #, A6, # SPECIFICATIONS ARE CHANGED TO ** :
  ^, /, 5X, ** THE #, A6, # MEAN SQUARE VALUES OF THIS#, 7X, ** #, /, 5X
  ^, ** ITERATION , THE COMPROMISE SOLUTION RESULTS ** )
90 FORMAT(8X, #OUTPUTS#, 8X, #MEAN SQUARE#, 3X, #NORMED#, 6X, #WEIGHT#,
  ^3X, ** #, 4X, #INPUTS#, 8X, #MEAN SQUARE#, 3X, #NORMED#, 6X, #WEIGHT#,
  ^, 5X, #NUMBER LABEL#, 7X, #VALUE#, 7X, #VALUE#, 7X, #VALUE#, 4X,
  ^# NUMBER LABEL#, 7X, #VALUE#, 7X, #VALUE#, 6X, # (RII) #)
91 FORMAT(6X, F3.0, 2X, A10, 1X, E11.5, 1X, E11.5, 1X, E11.5, # * #,
  ^F3.0, 2X, A10, 1X, E11.5, 1X, E11.5, 1X, E11.5)
92 FORMAT(6X, F3.0, 2X, A10, 1X, E11.5, 1X, E11.5, 1X, E11.5, # * #)
93 FORMAT(58X, ** #, F3.0, 2X, A10, 1X, E11.5, 1X, E11.5, 1X, E11.5)
94 FORMAT(35X, ** #, 45X, ** #, /, 28X, #YSUM# = E11.5,
  ^39X, #USUM# = #, E11.5, /, 5X, #TOTAL NORM SPEC# = #, E11.5)
95 FORMAT(SX, ** CONSTRAINED SPECIFICATIONS SOUGHT#, 11X, ** )
96 FORMAT(SX, ** CONSTRAINED SPECIFICATIONS FOUND#, 12X, ** )
99 FORMAT(SX, ** *****
  ^# ***** #, /, 5X, ** ALGORITHM CONVERGING VERY SLOWLY. THESE#
  ^, # MEAN SQUARE VALUES ** #, /, 5X, ** ARE GOOD APPROXIMATIONS TO#
  ^, # THE DESIRED SPECIFICATIONS.#, 5X, ** #, /, 5X, ** FOR A CLOSER #,
  ^# APPROXIMATION, THE ALGORITHM SHOULD BE#, 8X, ** #, /, 5X, ** RE#
  ^, # STARTED WITH THESE WEIGHTS AS THE INITIAL WEIGHTS AND#, 4X, **
  ^, /, 5X, ** THE CURRENT VALUES OF EITHER EY(I,2) OR EU(I,2) AS THE #

```

^,4X,##,/,5X,## UPDATE VECTOR,46X,##,/,5X,
 ^#####)

C##### INITIALIZE PARAMETERS #####

```

105 NYNU=NY+NU
    RNYNU=NYNU/1.
    MIN=NU
    IF(NY.LT.NU) MIN=NY
    NYU=NY-NU
    MAX=IABS(NYU)+MIN
    MINN=MIN+1
    DESCTN=1.0E+99
    II=0
    PWRS=0.
    DO 111 I=1,NY
      EYNORM(I,1)=1.0E+99
      EY(I,2)=0.
111  CONTINUE
      FLAGS(11)=.FALSE.
      SOLN=.FALSE.
      NSOLN=.FALSE.
      SSOLN=.FALSE.
      INTER=.TRUE.
      CONST=.F.
      RZFLG=.F.
      IF((FLAGS(4)).AND.(FLAGS(10))) CONST=.T.
      XHAT=.F.
      DO 112 I=1,NU
        EUNORM(I,1)=1.0E+99
        EU(I,2)=0.
112  CONTINUE
        NY2=NY*NY
        NU2=NU*NU
        N43=4*NY+3*NU
        N32=3*NY+2*NU
        NA2=NA*NA
        NAU=NA*NU
        N22=2*NY+2*NU
        N33=3*NY+3*NU
        N21=2*NY+NU
        N2=2*NY
        N2A=2*NA
        N2U=2*NU
        N3U=3*NU
        N4U=4*NU
        N5U=5*NU
        N6U=6*NU
        N2Z=2*NZ
        NST=N43+NA2+NAU
        NST1=NST+MAX*MAX
        SP=1.+EPS
        SPM=1.-EPS

```

C##### PARTIAL INPUT DATA ECHO CHECK #####

```

      IF(.NOT. FLAGS(1)) GO TO 110
      WRITE(6,1)
      WRITE(6,10)NU,NY,EPS
      DO 100 I=1,9

```

```

      WK(I)=SHFALSE
      IF(FLAG(I)) WK(I)=SH TRUE
100  CONTINUE
      WRITE(6,20)(WK(I),I=2,9)
      WRITE(6,30)
      DO 101 I=1,MIN
      WRITE(6,40)I,TITLY(I),SIGMA(I),I,TITLU(I),MU(I)
101  CONTINUE
      IF(NY.EQ.NU) GO TO 110
      IF(MIN.EQ.NY) GO TO 103
      DO 102 I=MINN,MAX
      WRITE(6,41)I,TITLY(I),SIGMA(I)
102  CONTINUE
      GO TO 110
103  DO 104 I=MINN,MAX
      WRITE(6,42)I,TITLU(I),MU(I)
104  CONTINUE

C***** CALCULATE STEADY STATE FILTER RICCATI *****

110  CALL MP31(NRM,NA,MM,NRU,NZ,UUI,NRM,NZ,NA,MM,NA,WK(N43+NA2+1))
      CALL MP32(NRD,NA,DD,NRW,NW,WRD,NW,NA,DD,NA,WK(N43+1))
      CALL POTTER(=FILTER=,NA,AA,NRA,WK(N43+NA2+1),NA,WK(N43+1),NA,PP
      ^,NRP,WK(N43+2*NA2+1),EIGAFM,WK(N43+6*NA2+1),.FALSE.,FLAG(3))
C***** SAVE PMTUIMP *****
      CALL MP3(NRP,NA,PP,NA,NA,WK(N43+NA2+1),NRP,NA,NA,PP,NA,
      ^WK(N43+1))
C*****CHECK*****
      IF(.NOT. FLAG(3)) GO TO 120:
      WRITE(6,1)
      WRITE(6,3)
      WRITE(6,50)GH-DP/DT
      WRITE(6,3)
      WRITE(6,2)
      CALL MPRINT(NA,WK(N43+2*NA2+1),NA,NA,10,S,=-DP/DT,0)

C***** FIXED COMPONENT (CPCT) CHECK *****

120  CALL MP32(NRC,NY,CC,NRP,NA,PP,NRC,NA,NY,CC,NY,WK(N43+NA2+1))
      NDUM=0
      DO 130 I=1,NY
      WK(N33+I)=WK(N43+NA2+(I-1)*NY+I)
      CPCTN=WK(N33+I)/SIGMA(I)
      IF(CPCTN.LT.SP) GO TO 130
      NDUM=NDUM+1
      IF(NDUM.NE.1) GO TO 131
      WRITE(6,1)
      WRITE(6,3)
131  WRITE(6,2)
      WRITE(6,60)I,TITLY(I)
130  CONTINUE
      IF (NDUM.EQ.0) GO TO 140
      NSOLN=.T.
      FLAG(2)=.T.
      WRITE(6,61)
      WRITE(6,3)
      WRITE(6,2)
      WRITE(6,62)
      DO 132 I=1,NY
      WRITE(6,63)I,TITLY(I),WK(N33+I),SIGMA(I)

```

132 CONTINUE

C***** SET INITIAL GUESS FOR Q AND R(INVERSE) *****

```

140 DO 142 I=1,NY
    DO 143 J=1,NY
        QQ(I,J)=0.
        IF(I.EQ.J) QQ(I,J)=1.0/SIGMA(I)
143 CONTINUE
142 CONTINUE
    DO 144 I=1,NU
        DO 145 J=1,NU
            RRI(I,J)=0.
            IF(I.EQ.J) RRI(I,I)=MU(I)
145 CONTINUE
144 CONTINUE
    IF(.NOT.FLAGS(4)) GO TO 150
    IF(FLAGS(2)) CALL MAREAD(NY,EY(1,2),NY,1,UPDATE NUM)
    IF(.NOT.FLAGS(2)) CALL MAREAD(NU,EU(1,2),NU,1,UPDATE NUM)
    CALL MAREAD(NY,WK(N22+1),NY,1,QQ INITIAL)
    CALL MAREAD(NU,WK(N32+1),NU,1,RR INITIAL)
    DO 146 I=1,NY
        QQ(I,I)=WK(N22+I)
146 CONTINUE
    DO 147 I=1,NU
        RRI(I,I)=WK(N32+I)**(-1)
147 CONTINUE

```

C***** CALCULATE STEADY STATE CONTROL RICCATI *****

C***** IDENTIFY ACTUATORS USING ONLY UNDISTURBABLE ESTIMATES *****

```

150 IF(.NOT.RZFLG) GO TO 151
    XHAT=.T.
    GO TO 162
151 CALL MP31(NRC,NA,CC,NRG,NY,QQ,NRC,NY,NA,CC,NA,WK(NST
    ^+NA2+1))
    CALL MP32(NRB,NA,BB,NRR,NU,RRI,NRB,NU,NA,BB,NA,WK(NST+1))
    CALL POTTER(=CONTROL,NA,AA,NRA,WK(NST+1),NA,WK(NST+NA2+1),NA,
    ^KK,NRK,WK(NST+2*NA2+1),EICABG,WK(NST+6*NA2+1),.FALSE.,FLAGS(5))
C*****CHECK*****
    IF(.NOT.FLAGS(5)) GO TO 170
    WRITE(6,1)
    WRITE(6,3)
    WRITE(6,50)GH-DK/DT
    WRITE(6,3)
    WRITE(6,2)
    CALL MPRINT(NA,WK(NST+2*NA2+1),NA,NA,10,5,=-DK/DT,0)

```

C*** IDENTIFY ACTUATORS WHICH EFFECT ONLY UNOBSERVABLE STATES***

```

170 CALL MULRTR(BB,KK,WK(N43+NA2+1),NU,NA,NA,NRB,NRK,NU)
    IF((I1.GT.0).AND.(.NOT.QZFLG)) GO TO 160
    CALL MULRRT(WK(N43+NA2+1),WK(N43+NA2+1),
    ^WK(NST+NA2+1),NU,NA,NU,NU,NU)
    BTKM=0.
    DO 171 I=1,NU
        IF(WK(NST+NA2+NU*(I-1)+I).GT.BTKM) BTKM=WK(NST+NA2
    ^+NU*(I-1)+I)
    WK(NST+NA2+NU2+I)=WK(NST+NA2+NU*(I-1)+I)
171 CONTINUE

```

```

162 NDUM=0
DO 172 J=1,NU
  I=NU+1-J
  IF(.NOT.XHAT) WK(N21+I)=TITLU(I)
  IF(.NOT.XHAT) WK(NST+NA2+NU2+I)=SQRT(WK(NST+NA2+NU2+I)/BTKM)
  IF((WK(NST+NA2+NU2+I).GT.EPS).AND.(.NOT.XHAT)) GO TO 172
  REPS=(1/RR1(I,I))*MU(I)
  IF((XHAT).AND.(REPS.GT.EPS)) GO TO 172
  NDUM=NDUM+1
  IF(NDUM.NE.1) GO TO 173
  WRITE(6,1)
  WRITE(6,3)
173 WRITE(6,70)I,TITLU(I)
  WRITE(6,2)
C***** DELETE ACTUATOR IF NECESSARY *****
  CNUM(1)=I
  RNUM(1)=I
  CALL MDROP(NRB,NA,NU,BB,0,RNUM,1,CNUM,0)
  CALL MDROP(NRW,NW,NW,WW,1,RNUM,1,CNUM,0)
  IF(NW.NE.NU) CALL MDROP(NRD,NA,NW,DD,0,RNUM,1,CNUM,0)
  CALL MDROP(NU,NU,1,TITLU,1,RNUM,0,CNUM,0)
  CALL MDROP(NU,NU,1,MU,1,RNUM,0,CNUM,0)
172 CONTINUE
  IF((NDUM.EQ.0).AND.(.NOT.XHAT)) GO TO 160
  NNU=NU
  NU=NU-NDUM
  NW=NW-NDUM
  IF(.NOT.XHAT) WRITE(6,71)
  IF(XHAT) WRITE(6,75)
  WRITE(6,3)
  IF(XHAT) GO TO 105
  WRITE(6,2)
  WRITE(6,73)
  DO 174 I=1,NNU
    WRITE(6,72)I,WK(N21+I),WK(NST+NA2+NU2+I)
174 CONTINUE
  GO TO 105

***** CALCULATE STEADY STATE VARIANCE OF STATE ESTIMATES *****
C***** FORM A+BG MATRIX *****
160 CALL MULT(WK(NST+1),KK,WK(NST+NA2+1),NA,NA,NA,NA,NRK,NA)
  CALL MADD(AA,WK(NST+NA2+1),WK(NST+1),NA,NA,NRA,NA,NA,-1)
C***** SOLVE LYAP EQUATION *****
  CALL GEIGEN(NA,WK(NST+1),NA,4,WK(NST+5*NA2+1),0,LOGHTS SUB#
  ^,EIGABG,NA,WK(NST+NA2+1),WK(NST+3*NA2+1),#A+BG#)
  CALL LYAP2(EIGABG,NA,WK(NST+NA2+1),WK(NST+3*NA2+1),NA,WK(N43+
  ^1),NRX,XXH,NA,WK(NST+6*NA2+1))
C*****CHECK*****
  IF(.NOT.FLAGS(5)) GO TO 180
  WRITE(6,1)
  WRITE(6,3)
  WRITE(6,50)GH-DX/DT
  WRITE(6,3)
  WRITE(6,2)
  CALL LYCHK(NA,NA,WK(NST+1),NRX,XXH,WK(NST+6*NA2+1),NA,
  ^WK(N43+1),2)

C***** CALCULATE MEAN SQUARE VALUES *****

```

C***** CALCULATE INPUT MEAN SQUARE VALUES AND CHECKS *****

```

180 NUSPEC=0
    NSTOP=0
    USUM=0.
    UUSUM=0.
    NUSUM=0
    NUMAX=0
    DESCYU=0.
    DESCYU=0.
    DUM=0.
    CALL MP32(NU,NU,WK(N43+NA2+1),NRX,NA,XXH,NU,NA,NU,
    WK(N43+NA2+1),NU,WK(NST+1))
    DO 183 I=1,NU
        EU(I)=WK(NST+NU*(I-1)+I)*RRI(I,I)**2
        EUNORM(I,2)=EU(I)/MU(I)
        IF(FLAG(2)) GO TO 195
        WK(NST1+I)=EU(I,2)
        IF(.NOT.CONST) EU(I,2)=0.
        IF(EUNORM(I,2).LE.SP) EU(I,2)=1.
        NUSUM=NUSUM+INT(EU(I,2))
195 IF(EUNORM(I,2).LE.SP) NUSPEC=NUSPEC+1
        IF(EUNORM(I,2).GE.SPM) NUMAX=NUMAX+1
        IF(EUNORM(I,2).GT.SP) DUM=(EUNORM(I,2)-1.)
        IF(DUM.GT.DESCTU) DESCYU=DUM
        USUM=USUM+EUNORM(I,2)
        UUSUM=(EU(I)/RRI(I,I))+UUSUM
        DEL=ABS(EUNORM(I,2)-EUNORM(I,1))
        IF(DEL.LT.EPS) NSTOP=NSTOP+1
        IF(FLAG(2)) GO TO 183
        DUM=0.
        IF((EU(I,2).GT.SPM).AND.(EUNORM(I,2).GT.SP)) DUM=EUNORM(I,2)-1.
        IF(DUM.GT.DESCYU) DESCYU=DUM
183 CONTINUE
    NUBSPC=NU-NUMAX
    NUOSPC=NU-NUSPEC
    NUISPC=NU-NUBSPC-NUOSPC

```

C***** CALCULATE OUTPUT MEAN SQUARE VALUES AND CHECKS *****

```

    NYMAX=0
    NOZERO=0
    DESCYU=0.
    NYSPEC=0
    YSUM=0.
    NYSUM=0
    CALL MP32(NRC,NY,CC,NRX,NA,XXH,NRC,NA,NY,CC,NY,WK(NST+1))
    DO 184 I=1,NY
        EY(I)=WK(N33+I)+WK(NST+NY*(I-1)+I)
        EYNORM(I,2)=EY(I)/SIGMA(I)
        IF(.NOT.FLAG(2)) GO TO 191
        WK(NST1+I)=EY(I,2)
        IF(.NOT.CONST) EY(I,2)=0.
        IF(EYNORM(I,2).LE.SP) EY(I,2)=1.
        NYSUM=NYSUM+INT(EY(I,2))
191 IF(EYNORM(I,2).LE.SP) NYSPEC=NYSPEC+1
        IF(EYNORM(I,2).GE.SPM) NYMAX=NYMAX+1
        QEPS=QQ(I,I)*SIGMA(I)
        IF((QEPS.LE.EPS).AND.(EYNORM(I,2).LT.SPM)) NOZERO=NOZERO+1
        DUM=0.
        IF((EYNORM(I,2).GT.SP).AND.(QEPS.GT.EPS)) DUM=(EYNORM(I,2)-1.)
        IF(DUM.GT.DESCTY) DESCYU=DUM
        YSUM=YSUM+EYNORM(I,2)

```

```

DEL=ABS(EYNORM(I,2)-EYNORM(I,1))
IF(DEL.LT.EPS) NSTOP=NSTOP+1
IF(.NOT. FLAGS(2)) GO TO 184
DUM=0.
IF((EY(I,2).GT.SPM).AND.(EYNORM(I,2).GT.SP)) DUM=EYNORM(I,2)-1.
IF(DUM.GT.DESCYU) DESCYU=DUM
184 CONTINUE
TSPEC=YSUM+USUM
NYBSPC=NY-NYMAX
NYOSPC=NY-NYSPEC
NYISPC=NY-NYBSPC-NYOSPC
NYACTU=NY-NQZERO
IF((CONST).AND.(FLAGS(2))) NYSUM=NYSUM-NQZERO
IF((NUISPC.EQ.NU).AND.(FLAGS(2))) CONST=.T.
IF((NYISPC.EQ.NYACTU).AND.(.NOT.FLAGS(2))) CONST=.T.

```

C***** CALCULATE DESCENT FUNCTION *****

```

DESCTO=DESCTN
DESCTN=DESCTU
IF(.NOT. FLAGS(2)) DESCYN=DESCTY
IF((DESCYU.GT.DESCTN).AND.(CONST)) DESCYN=DESCYU
IF(DESCTN.EQ.0.) GO TO 186
IF(FLAGS(11)) GO TO 186
IF(DESCTN.GT.DESCTO) GO TO 187
GO TO 186

```

C***** RESET CALCULATIONS *****

```

187 IF(PWR.EQ..5) GO TO 175
IF((.NOT.FLAGS(2)).AND.(.NOT.CONST).AND.(PWR.EQ.1.)) GO TO 175
IF(DESCTO.EQ.0) GO TO 175
DESCTN=DESCTO
DO 188 I=1,NY
EYNORM(I,2)=EYNORM(I,1)
QD(I,1)=WK(N22+I)
IF(FLAGS(2)) EY(I,2)=WK(NST1+I)
188 CONTINUE
DO 189 I=1,NU
EUNORM(I,2)=EUNORM(I,1)
RRI(I,1)=1.0/WK(N32+I)
IF(.NOT.FLAGS(2)) EU(I,2)=WK(NST1+I)
189 CONTINUE
PWR=0.
GO TO 230
175 PWR=0.

```

C***** SOLUTION CHECKS *****

```

186 IF(DESCTN.LT.DESCTO) FLAGS(11)=.F.
IF(NSOLN) GO TO 190
IF((.NOT.FLAGS(4)).AND.(TSPEC.GE.RMYNU).AND.(II.EQ.0)) NSOLN=.T.
IF((NYSPEC.NE.NY).OR.(NUSPEC.NE.NU)) GO TO 190
SOLN=.T.
GO TO 197
190 IF(FLAGS(2)) GO TO 192
IF((NYISPC.NE.NYACTU).OR.(NUISPC.NE.NUSUM)) GO TO 193
GO TO 194
192 IF((NUISPC.NE.NU).OR.(NYISPC.NE.NYSUM)) GO TO 193
194 NSOLN=.T.
SSOLN=.T.
GO TO 197

```

```

193 IF(II.GE.ITNUM) GO TO 197
   IF(NSTOP.NE.NYNU) GO TO 203
   FLAGS(11)=.T.
   GO TO 203
197 INTER=.FALSE.
   IF(.NOT. FLAGS(9)) GO TO 210
   CALL MWRITE(NY,EY,NY,1,MY MS VALS #)
   CALL MWRITE(NU,EU,NU,1,U MS VALS #)
   IF(FLAGS(2)) CALL MWRITE(NY,EY(1,2),NY,1,UPDATE NUM#)
   IF(.NOT.FLAGS(2)) CALL MWRITE(NU,EU(1,2),NU,1,UPDATE NUM#)

C***** STORE RESULTS *****

203 DO 200 I=1,NY
   WK(N22+I)=QQ(I,I)
200 CONTINUE
   DO 201 I=1,NU
   WK(N32+I)=1.0/RRI(I,I)
201 CONTINUE
   IF((.NOT.FLAGS(7)).AND.(INTER)) GO TO 210
   CALL MWRITE(NY,WK(N22+1),NY,1,OUTPUT HTS#)
   CALL MWRITE(NU,WK(N32+1),NU,1,INPUT HTS #)
   CALL MWRITE(NY,EYNORM(1,2),NY,1,Y NORM VAL#)
   CALL MWRITE(NU,EUNORM(1,2),NU,1,U NORM VAL#)
   CALL MWRITE(N2A,EIGABG,N2A,1,A+BG EIGS#)
   IF (INTER) GO TO 210
   CALL MWRITE(N2A,EIGAFH,N2A,1,A-FM EIGS#)
C***** SAVE CURRENT VALUES *****
210 DO 181 I=1,NY
   WK(I)=I
   WK(I+NY)=TITLY(I)
   EYNORM(I,1)=EYNORM(I,2)
   WK(NST+I)=QQ(I,I)
181 CONTINUE
   DO 182 I=1,NU
   WK(N2+I)=I
   WK(N21+I)=TITLU(I)
   EUNORM(I,1)=EUNORM(I,2)
   WK(NST+NY+I)=1./RRI(I,I)
182 CONTINUE

C***** PRINT RESULTS *****

   IF((.NOT.FLAGS(8)).AND.(INTER)) GO TO 230
   IF((.NOT.FLAGS(8)).AND.(.NOT.INTER)) GO TO 250
C***** INPUT ORDERING *****
   DO 185 I=1,NU
   NP=NU+1-I
   DO 185 J=1,NP
   IF(J.EQ.NP) GO TO 185
   IF(EUNORM(J,2).GE.EUNORM(J+1,2)) GO TO 185
   TEMP1=EUNORM(J,2)
   TEMP2=EU(J)
   TEMP3=WK(N2+J)
   TEMP4=WK(N21+J)
   TEMP5=WK(NST+NY+J)
   EUNORM(J,2)=EUNORM(J+1,2)
   EU(J)=EU(J+1)
   WK(N2+J)=WK(N2+J+1)
   WK(N21+J)=WK(N21+J+1)

```

```

      WK(NST+NY+J)=WK(NST+NY+J+1)
      EYNORM(J+1,2)=TEMP1
      EU(J+1)=TEMP2
      WK(N2+J+1)=TEMP3
      WK(N21+J+1)=TEMP4
      WK(NST+NY+J+1)=TEMP5
185  CONTINUE
C***** OUTPUT ORDERING *****
      DO 208 I=1,NY
      NP=NY+1-I
      DO 208 J=1,NP
      IF(J.EQ.NP) GO TO 208
      IF(EYNORM(J,2).GE.EYNORM(J+1,2)) GO TO 208
      TEMP1=EYNORM(J,2)
      TEMP2=EY(J)
      TEMP3=WK(J)
      TEMP4=WK(NY+J)
      TEMP5=WK(NST+J)
      EYNORM(J,2)=EYNORM(J+1,2)
      EY(J)=EY(J+1)
      WK(J)=WK(J+1)
      WK(J+NY)=WK(J+1+NY)
      WK(NST+J)=WK(NST+J+1)
      EYNORM(J+1,2)=TEMP1
      EY(J+1)=TEMP2
      WK(J+1)=TEMP3
      WK(NY+J+1)=TEMP4
      WK(NST+J+1)=TEMP5
208  CONTINUE
C***** PRINT *****
      WRITE(6,1)
      WRITE(6,3)
      WRITE(6,80)II
      IF(II.EQ.0) GO TO 214
      IF(FLAGS(10)) GO TO 216
      IF(FLAGS(2)) WRITE(6,81) R PWR
      IF(.NOT.FLAGS(2)) WRITE(6,81) Q PWR
      GO TO 217
214  WRITE(6,74)
      IF(.NOT.FLAGS(4)) WRITE(6,89)SYSTEM,GIVEN
      GO TO 217
216  WRITE(6,81)Q AND R,PWR
217  IF(.NOT.SOLN) GO TO 211
      WRITE(6,82)
      WRITE(6,3)
      GO TO 218
211  IF(.NOT.NSOLN) GO TO 213
      WRITE(6,83)
      IF(.NOT.SSOLN) GO TO 221
      WRITE(6,96)
      IF(FLAGS(2)) WRITE(6,87)OUTPUT,OUTPUT,INPUT
      IF(.NOT.FLAGS(2)) WRITE(6,87)INPUT,INPUT,OUTPUT
      WRITE(6,3)
      GO TO 218
215  WRITE(6,88)SYSTEM,GIVEN
      WRITE(6,3)
      WRITE(6,99)
      GO TO 218
221  WRITE(6,95)
      IF(II.GE.ITNUM) GO TO 215

```

```

WRITE(6,3)
GO TO 218
213 WRITE(6,84)
IF(II.GE.ITNUM) GO TO 215
WRITE(6,3)
218 WRITE(6,2)
WRITE(6,90)
WRITE(6,2)
DO 219 I=1,MIN
WRITE(6,91)WK(I),WK(NY+I),EY(I),EYNORM(I,2),WK(NST+I),
~WK(N2+I),WK(N21+I),EU(I),EUNORM(I,2),WK(NST+NY+I)
219 CONTINUE
IF(NY.EQ.NU) GO TO 222
DO 220 I=MINN,MAX
IF(MIN.EQ.NU) WRITE(6,92)WK(I),WK(NY+I),EY(I),EYNORM(I,2),
~WK(NST+I)
IF(MIN.EQ.NY) WRITE(6,93)WK(N2+I),WK(N21+I),EU(I),EUNORM(I,2)
~,WK(NST+NY+I)
220 CONTINUE
222 WRITE(6,94)YSUM,USUM,TSPEC
WRITE(6,1)
WRITE(6,3)
WRITE(6,51)A+BG.
WRITE(6,3)
WRITE(6,2)
WRITE(6,53)
I1=0
DO 223 I=1,N2A,2
I1=I1+1
FREQ=SQRT(EIGABG(I)**2+EIGABG(I+1)**2)
IF(FREQ.EQ.EIGABG(I)) FREQ=0.
DAMP=1
IF(FREQ.NE.0.) DAMP=ABS(EIGABG(I))/FREQ
WRITE(6,54)I1,EIGABG(I),EIGABG(I+1),FREQ,DAMP
225 CONTINUE
IF(INTER) GO TO 230
WRITE(6,2)
WRITE(6,3)
WRITE(6,51)A-FM.
WRITE(6,52)
WRITE(6,3)
WRITE(6,2)
WRITE(6,53)
I1=0
DO 226 I=1,N2A,2
I1=I1+1
FREQ=SQRT(EIGAFM(I)**2+EIGAFM(I+1)**2)
IF(FREQ.EQ.EIGAFM(I)) FREQ=0.
DAMP=1
IF(FREQ.NE.0.) DAMP=ABS(EIGAFM(I))/FREQ
WRITE(6,54)I1,EIGAFM(I),EIGAFM(I+1),FREQ,DAMP
226 CONTINUE
GO TO 250

```

C***** UPDATE EQUATIONS *****

```

230 II=II+1
FLAGS(10)=.F.
QZFLG=.F.
RZFLG=.F.

```

```

      PURS=PURS+1.
      PUR=1.
      IF((PURS.LE.SP).AND.(PURS.GE.SPM).AND.((FLAGS(2)).OR.(CONST)))
        PUR=.5
      IF(PURS.GT.SP) PUR=PURS-1.
      IF(.NOT.FLAGS(2)) GO TO 231
C***** INPUT CONSTRAINED UPDATES *****
      DO 240 I=1,NU
      RRI(I,I)=EUNORM(I,1)**(-PUR)*RRI(I,I)
      REPS=(1/RRI(I,I))*MU(I)
      IF(REPS.LT.EPS) RZFLG=.T.
240  CONTINUE
      IF(NYBSPC.EQ.0) GO TO 150
      IF(.NOT.CONST) GO TO 150
      FLAGS(10)=.TRUE.
      DO 233 I=1,NY
      IF(QQ(I,I).EQ.0.) GO TO 233
      IF(EY(I,2).GT.SPM) QQ(I,I)=EYNORM(I,1)**PUR*QQ(I,I)
      QEPS=QQ(I,I)*SIGMA(I)
      IF((QEPS.GT.EPS).OR.(EYNORM(I,1).GE.SPM)) GO TO 233
      QQ(I,I)=0.
      QZFLG=.T.
233  CONTINUE
      GO TO 150
C***** OUTPUT CONSTRAINED UPDATES *****
231  DO 242 I=1,NY
      IF(QQ(I,I).EQ.0.) GO TO 242
      QQ(I,I)=EYNORM(I,1)**PUR*QQ(I,I)
      QEPS=QQ(I,I)*SIGMA(I)
      IF((QEPS.GT.EPS).OR.(EYNORM(I,1).GE.SPM)) GO TO 242
      QQ(I,I)=0.
      QZFLG=.T.
242  CONTINUE
      IF(NUBSPC.EQ.0) GO TO 150
      IF(.NOT.CONST) GO TO 150
      FLAGS(10)=.T.
      DO 235 I=1,NU
      IF(EU(I,2).GT.SPM) RRI(I,I)=EUNORM(I,1)**(-PUR)*RRI(I,I)
      REPS=(1/RRI(I,I))*MU(I)
      IF(REPS.LT.EPS) RZFLG=.T.
235  CONTINUE
      GO TO 150
250  CONTINUE

C***** ACTUATOR / SENSOR EFFECTIVENESS CALCULATIONS *****
C***** SET UP A-FM *****
      CALL MP31(NRM,NA,MM,NRU,NZ,UUI,NRM,NZ,NA,MM,NA,WK(NST+1))
      CALL MULT(PP,WK(NST+1),WK(NST+NA2+1),NA,NA,NA,NRP,NA,NA)
      CALL MADD(AA,WK(NST+NA2+1),WK(NST+1),NA,NA,NRA,NA,NA,-1)
C***** SOLVE L LYAP EQ. *****
      CALL MP31(MU,NA,WK(N43+NA2+1),NRR,MU,RRI,MU,MU,NA,WK(N43+NA2+1),
        NA,WK(NST+2*NA2+1))
      CALL GEIGEN(NA,WK(NST+1),NA,4,WK(NST+7*NA2+1),0,ALQGSAS,EIGAFM,
        NA,WK(NST+3*NA2+1),WK(NST+5*NA2+1),A-FM)
      CALL LYAP1(EIGAFM,NA,WK(NST+3*NA2+1),WK(NST+5*NA2+1),NA,
        WK(NST+2*NA2+1),NA,WK(NST+NA2+1),NA,WK(NST+7*NA2+1))
C***** CALCULATE ACTUATOR NOISE CONTRIBUTION *****
      CALL MADD(KK,WK(NST+NA2+1),WK(NST+2*NA2+1),NA,NA,NA,NA,
        NA,1)

```

```

CALL MP31(NRB, NU, BB, NA, NA, WK(NST+2*NA2+1), NRB, NA, NU, BB, NU,
~WK(NST+3*NA2+NU2+1))
CALL MULT(WK(NST+3*NA2+NU2+1), WW, WK(NST+3*NA2+1), NU, NU, NU, NU, NRW
~, NU)
DO 260 I=1, NU
UU=EUNORM(I, 1)*MU(I)*(1./RRI(I, I))
WK(I)=UU-WK(NST+3*NA2+(I-1)*NU+I)
IF(FLAGS(2)) WK(I)=(UU/
~WK(NST+3*NA2+(I-1)*NU+I))*UUSUM/(UUSUM-UU)
WK(I+NU)=WK(NST+3*NA2+(I-1)*NU+I)
WK(I+N2U)=I
WK(I+N3U)=TITLU(I)
WK(I+N4U)=I
WK(I+NSU)=TITLU(I)
260 CONTINUE
C***** CALCULATE SENSOR RANKING *****
CALL MP3(NRP, NA, PP, NA, NA, WK(NST+NA2+1), NRP, NA, NA, PP, NA,
~WK(NST+3*NA2+NU2+1))
CALL MP32(NRM, NZ, MM, NA, NA, WK(NST+3*NA2+NU2+1), NRM, NA, NZ,
~MM, NZ, WK(NST+4*NA2+NU2+NZ*NZ+1))
CALL MULT(UUI, WK(NST+4*NA2+NU2+NZ*NZ+1), WK(NST+4*NA2+NU2+1),
~NZ, NZ, NZ, NRW, NZ, NZ)
DO 261 I=1, NZ
WK(I+NSU)=WK(NST+4*NA2+NU2+(I-1)*NZ+I)
WK(I+NSU+NZ)=I
WK(I+NSU+N2Z)=TITLZ(I)
261 CONTINUE
C***** PRINT RESULTS (ORDERED) *****
WRITE(6, 1)
MIN=NU
IF(NZ.LT.NU) MIN=NZ
NZU=NZ-NU
MAX=IABS(NZU)+MIN
MINN=MIN+1
C***** ACTUATOR ORDERING *****
DO 262 I=1, NU
NP=NU+1-I
DO 262 J=1, NP
IF(J.EQ.NP) GO TO 262
IF(WK(J).GE.WK(J+1)) GO TO 266
TP1=WK(J)
TP2=WK(J+N2U)
TP3=WK(J+N3U)
WK(J)=WK(J+1)
WK(J+N2U)=WK(J+1+N2U)
WK(J+N3U)=WK(J+1+N3U)
WK(J+1)=TP1
WK(J+1+N2U)=TP2
WK(J+1+N3U)=TP3
266 IF(WK(J+NU).GE.WK(J+1+NU)) GO TO 262
TP1=WK(J+NU)
TP2=WK(J+N4U)
TP3=WK(J+NSU)
WK(J+NU)=WK(J+1+NU)
WK(J+N4U)=WK(J+1+N4U)
WK(J+NSU)=WK(J+1+NSU)
WK(J+1+NU)=TP1
WK(J+1+N4U)=TP2
WK(J+1+NSU)=TP3
262 CONTINUE

```

```

C*****: SENSOR ORDERING *****
DO 263 I=1,NZ
NP=NZ+1-I
DO 263 J=1,NP
IF(J.EQ.NP) GO TO 263
IF(WK(NSU+J).GE.WK(NSU+J+1)) GO TO 263
TP1=WK(NSU+J)
TP2=WK(NSU+NZ+J)
TP3=WK(NSU+N2Z+J)
WK(NSU+J)=WK(NSU+J+1)
WK(NSU+NZ+J)=WK(NSU+NZ+J+1)
WK(NSU+N2Z+J)=WK(NSU+N2Z+J+1)
WK(NSU+J+1)=TP1
WK(NSU+NZ+J+1)=TP2
WK(NSU+N2Z+J+1)=TP3
263 CONTINUE
C***** PRINT *****
WRITE(6,15)
WRITE(6,11)
WRITE(6,2)
DO 264 I=1,MIN
WRITE(6,12)WK(NSU+NZ+I),WK(NSU+N2Z+I),WK(NSU+I),WK(N4U+I),
WK(NSU+I),WK(NU+I),WK(N2U+I),WK(N3U+I),WK(I)
264 CONTINUE
IF(NU.EQ.NZ) GO TO 269
DO 265 I=MIN,MAX
IF(MIN.EQ.NU) WRITE(6,13)WK(NSU+NZ+I),WK(NSU+N2Z+I),WK(NSU+I)
IF(MIN.EQ.NZ)
WRITE(6,14)WK(N4U+I),WK(NSU+I),WK(NU+I),WK(N2U+I),WK(N3U+I),WK(I)
265 CONTINUE
269 RETURN
END

SUBROUTINE MWRITE(NROW,MATRIX,NR,NC,NAME)
REAL MATRIX(NROW,NC)
1 FORMAT(1000 THE MATRIX #,A10,# (#,I3,# BY#,I3,#) #)
2 FORMAT(6E12.5)
WRITE(8,1)NAME,NR,NC
WRITE(8,2)((MATRIX(I,J),I=1,NR),J=1,NC)
RETURN
END

SUBROUTINE MAREAD(NROW,MATRIX,NR,NC,NAME)
REAL MATRIX(NROW,NC)
1 FORMAT(17X,A10,2X,I3,3X,I3)
2 FORMAT(6E12.5)
READ(5,1)NAME,NR,NC
READ(5,2)((MATRIX(I,J),I=1,NR),J=1,NC)
RETURN
END

SUBROUTINE LYCHK(NRA,NA,AA,NRX,XX,WK,NRM,MM,FLAG)
REAL AA(NRA,1),XX(NRX,1),WK(1),MM(NRM,1)
INTEGER NRA,N,NRX,FLAG,NRM
IF (FLAG.EQ.2) GO TO 100
CALL MULTR(AA,XX,WK,NA,NA,NRA,NRX,NA)
CALL MULT(XX,AA,WK(NA*NA+1),NA,NA,NA,NRX,NRA,NA)
120 CALL MADD(WK,WK(NA*NA+1),WK(2*NA*NA+1),NA,NA,NA,NA,NA,1)
CALL MADD(WK(2*NA*NA+1),MM,WK,NA,NA,NA,NRM,NA,1)
CALL MPRINT(NA,WK,NA,NA,9,5,LYCHECK#,0)

```

```

      GO TO 110
100 CALL MULT(AA,XX,UK,NA,NA,NA,NRA,NRX,NA)
   CALL MULVRT(XX,AA,UK(NA=NA+1),NA,NA,NA,NRX,NRA,NA)
      GO TO 120
110 RETURN
   END

   SUBROUTINE MDROP(NROWA,NRA,NCA,AA,NROWS,ROWNUM,NCOLS,
+COLNUM,FLAG)
   REAL AA(NROWA,NCA)
   INTEGER ROWNUM(NROWS),COLNUM(NCOLS)
   IF (NROWS.EQ.0) GO TO 101
   DO 100 J=1,NROWS
     NR=NRA-ROWNUM(J)
     IF(NR.EQ.0) GO TO 104
     CALL MEQ(AA(ROWNUM(J)+1,1),AA(ROWNUM(J),1),NR,NCA,NROWA,NROWA)
104  DO 100 I=1,NCA
     AA(NRA,I)=0.
100  CONTINUE
101  IF (NCOLS.EQ.0) GO TO 102
     DO 110 J=1,NCOLS
       NC=NCA-COLNUM(J)
       IF(NC.EQ.0) GO TO 105
       CALL MEQ(AA(1,COLNUM(J)+1),AA(1,COLNUM(J)),NRA,NC,NROWA,NROWA)
105  DO 110 I=1,NRA
     AA(I,NC)=0.
110  CONTINUE
102  IF (FLAG.EQ.0) GO TO 103
     CALL MPRINT(NROWA,AA,NRA,NCA,9,5,AA-TRUNC,0)
103  CONTINUE
     RETURN
   END

```

Note: For the LQGWTS subroutine just listed to work as advertised in Chapter 6, (σ^2, u^2) must be chosen such that no LQG controller can satisfy them. Currently, if a solution exists, the program will stop when any solution is found, (i.e. It will *not* search for the input-constrained or output-constrained solution in this case.) Also, the program currently does not check for $q_i \rightarrow \infty$, so uncontrollable systems could cause problems for the output-constrained search, (i.e. footnote on page 115).

Program LQHOOP

```

15599, T10, L10000, TU100000, MF140000, TC50, T1024
PFILES(GET, LSLIB3, ID=FME)
PFILES(GET, LSLIB2, ID=T1Y)
PFILES(GET, HOOPOO, ID=EJM)
RFL/120000
FTN4(MAN, R=0)
GET(LSLIB3, LBD)REL/BLS1
LOAD, LBD.
LOAD, LGO, LSLIB2, LSLIB3.
EXECUTE, , HOOPOO
PFILES(PUT, HDAT, GETP=U, X=TAPE8)

```

```
PROGRAM LQHOOP(INPUT, OUTPUT, TAPES=INPUT, TAPES=OUTPUT, TAPE8)
```

```
EXTERNAL MDROP, LYCHK, MWRITE, MAREAD
```

```

REAL AA(26, 26), BB(26, 24), CC(27, 26), WW(24, 24), UUI(39, 39),
IMM(39, 26), DD(27, 27), RRI(24, 24), SIGMA(24), MU(24), TITLY(24),
ITITLU(24), EY(24, 2), EU(24, 2), EYNORM(24, 2), EUNORM(24, 2), PP(26,
126), KK(26, 26), XXH(26, 26), WK(8332), EIGABC(104), EIGAFM(104),
ITITLZ(39)

```

```
INTEGER NROWS(3), RNUM(3), RNUM1(6), RNUM2(27)
```

```
LOGICAL ITT, FLAGS(11)
```

```

COMMON/MATIO/FORM, TOL
COMMON/HDRMPR/OSUB(2), KPAGE
COMMON/HEAD/TITLE(7), CASE(4), LINE

```

```

DATA RNUM/15, 12, 9/
DATA RNUM1/11, 10, 9, 8, 7, 2/
DATA RNUM2/39, 36, 33, 27, 24, 21, 18, 15, 14, 13, 12, 9, 6, 5, 4, 3, 2,
~1, 11, 10, 9, 8, 7, 6, 4, 3, 1/
DATA NROWS/3, 5, 18/

```

```
20 FORMAT(6(1X, E11.5))
```

```
30 FORMAT(3(1X, E11.5))
```

```
C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
```

```
C PAGE HEADER SET UP:
```

```
C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
```

```
FORM=0.
```

```
TOL=0.
```

```
LINE=1
```

```
CASE(1)=JPL HOOP: A=
```

```
CASE(2)=ANTENNA MOD=
```

```
CASE(3)=EL (13 MOD=
```

```
CASE(4)=ES)
```

```
OSUB(1)=LOG WT SPE=
```

```
OSUB(2)=CIFICATION=
```

```
KPAGE=0
```

```

C
C
C

```

```
MODEL SIZING
```

```
NRA=26
```

```
NRS=26
```

```
NRU=39
```

```
NRJ=24
```

```
NRK=24
```

```
NRK=26
```

```

      N=26
      NU=12
      NM=24
      NZ=39
      NRM=39
      NRP=26
      NRC=27
      NRQ=27
      NRX=26
      NY=27
C:.....
C      INITIALIZE MATRICIES
C:.....
      CALL MZERO(AA,N,N,NRA)
      CALL MZERO(BB,N,N,NRB)
      CALL MZERO(CC,NY,N,NRC)
      CALL MZERO(MM,NM,NM,NRM)
      CALL MZERO(UU,NZ,NZ,NRU)
      CALL MZERO(MM,NZ,N,NRM)
C:.....
C      SET UP AA MATRIX
C:.....
      READ(5,20)(AA(10+I,I),I=1,6)
      READ(5,20)(AA(10+I,I),I=7,10)
      READ(5,20)(AA(10+I,10+I),I=1,6)
      READ(5,20)(AA(10+I,10+I),I=7,10)
      CALL IDENT(AA(1,1),10,26)
      CALL IDENT(AA(21,24),3,26)
C      CALL MPRINT(NRA,AA,N,N,7,5,AA=0)
C:.....
C      SET UP BB MATRIX
C:.....
      DO 800 I=1,10
      READ(5,20)(BB(10+I,J),J=1,6)
      READ(5,20)(BB(10+I,J),J=7,12)
800  CONTINUE
      DO 830 I=1,3
      READ(5,20)(BB(23+I,J),J=1,6)
      READ(5,20)(BB(23+I,J),J=7,12)
830  CONTINUE
      CALL MEQ(BB,BB(1,13),N,12,NRB,NRB)
C      CALL MPRINT(NRB,BB,N,NU,6,5,BB=0)
C:.....
C      SET UP CC MATRIX
C:.....
      DO 810 I=1,27
      READ(5,20)(CC(I,J),J=1,6)
      READ(5,20)(CC(I,J),J=7,10)
      READ(5,30)(CC(I,J),J=21,23)
810  CONTINUE
      CALL MDROP(NRC,NY,N,CC,NROWS(1),RNUM,0,CNUM,0)
      NY=NY-3
C      CALL MPRINT(NRC,CC,NY,N,9,5,CC=0)
C:.....
C      SET MM MATRIX
C:.....
      DO 820 I=1,39
      READ(5,20)(MM(I,J),J=1,6)
      READ(5,20)(MM(I,J),J=7,12)
      READ(5,20)(MM(I,J),J=13,18)

```

```

      READ(5,20)(MM(I,J),J=19,20)
      READ(5,20)(MM(I,J),J=21,26)
820  CONTINUE
C    CALL MPRINT(NRM,MM,NZ,N,6,5,#MM#,0)
C:::
C    SET UP UU MATRIX
C:::
      READ(5,20)(UI(I,I),I=1,6)
      READ(5,20)(UI(I,I),I=7,12)
      READ(5,20)(UI(I,I),I=13,18)
      READ(5,20)(UI(I,I),I=19,24)
      READ(5,20)(UI(I,I),I=25,30)
      READ(5,20)(UI(I,I),I=31,36)
      READ(5,20)(UI(I,I),I=37,39)
      DO 821 I=1,39
      UI(I,I)=1.0/UI(I,I)
821  CONTINUE
C    CALL MPRINT(NRU,UI,NZ,NZ,6,5,#UI#,0)
C
C    SPECIFY OUTPUT AND CONTROL MAX VALUES
C
      DO 50 I=1,6
      SIGMA(I)=.000000012184697
50  CONTINUE
      DO 51 I=7,24
      SIGMA(I)=.0000000025
51  CONTINUE
      DO 52 I=1,NU
      MU(I)=.000000000001
52  CONTINUE
      ITT=.FALSE.
      IF(.NOT.ITT) GO TO 444
      CALL MAREAD(NY,SIGMA,NY,1,#Y MS UALS #)
      CALL MAREAD(NU,MU,NU,1,#U MS UALS #)
444  CONTINUE
C:::
C    SET UP WW MATRIX
C:::
      DO 701 I=1,NU
      WW(I,I)=.000000001
701  CONTINUE
      DO 702 I=1,12
      WW(NU+I,NU+I)=1.0E-13
702  CONTINUE
C    CALL MPRINT(NRW,WW,NU,NU,5,5,#WW#,0)
C
C    ACTUATOR/SENSOR LABELING
C
      TITLU(1)=10H  TX2
      TITLU(2)=10H  TY2
      TITLU(3)=10H  TZ2
      TITLU(4)=10H  TX6
      TITLU(5)=10H  TY6
      TITLU(6)=10H  TZ6
      TITLU(7)=10H  TX9
      TITLU(8)=10H  TY9
      TITLU(9)=10H  TZ9
      TITLU(10)=10H TX10
      TITLU(11)=10H TY10
      TITLU(12)=10H TZ10

```



```

CALL MDROP(NRM,NZ,N,MM,NROWS(3),RNUM2,0,RNUM2,0)
CALL MDROP(NRU,NZ,NZ,UUI,NROWS(3),RNUM2,NROWS(3),RNUM2,0)
CALL MDROP(NZ,NZ,1,TITLZ,NROWS(3),RNUM2,0,RNUM2,0)
NZ=NZ-NROWS(3)
C:.....
C          SET FLAGS FOR LOGITS
C:.....
  FLAGS(1)=.F.
  FLAGS(2)=.T.
  FLAGS(3)=.F.
  FLAGS(4)=.F.
  FLAGS(5)=.F.
  FLAGS(6)=.F.
  FLAGS(7)=.F.
  FLAGS(8)=.TRUE.
  FLAGS(9)=.F.
  FLAGS(10)=.F.
  EPS=.001
C:.....
C          SUBROUTINE LOGITS (TA DALLLLLLLL)
C:.....
  CALL LOGITS(NRA,N,AA,NRB,BB,NRB,NH,BB,NRH,WH,NRU,UUI,NRC,CC,NRM,
  IMM,SIGMA,MU,TITLY,TITLU,TITLZ,NY,NU,NZ,FLAGS,EY,EU,EYNORM,EUNORM,
  JNRP,PP,NRK,KK,NRX,XXH,EIGAFH,EIGABG,NRQ,QQ,NRR,ARI,WK,EPS,20)
  END

```

Program LQTELE

```

15599, TWO, L10000, TU100000, TC310, T1024, CM40000, MF150000
PFILES(GET, LSLIB3, ID=FME)
PFILES(GET, LSLIB2, ID=T1Y)
PFILES(GET, LOCMO, ID=TWO)
FTN4(MAN, R=0)
GET(LSLIB3, LBD)REL/BLS1
LOAD, LBD.
LOAD, LGO, LSLIB2, LSLIB3.
EXECUTE, , LOCMO
PFILES(PUT, TDATA, GETP=U, X=TAPEB)
PROGRAM LQTELE(INPUT, OUTPUT, TAPES=INPUT, TAPES=OUTPUT, TAPES)
EXTERNAL MDROP, LYCHK, MWRITE, MAREAD
REAL AA(24, 24), BB(24, 21), CC(3, 24), MM(23, 23), DD(24, 23)
~, UUI(45, 45), MM(45, 24)
REAL DD(3, 3), RRI(21, 21), SIG(3), MU(21), TITLY(3), TITLU(21),
~EY(3, 2), EU(21, 2), EYNORM(3, 2), EUNORM(21, 2), PP(24, 24), KK(24, 24)
~, XXH(24, 24), LK(7950), EABG(96), EAFM(96), TITLZ(45)
LOGICAL ITT, FLAGS(11)
INTEGER NROWS(2), RNUM1(9), RNUM2(33)
5 FORMAT(5(E13.5))
1 FORMAT((E13.5))
4 FORMAT(4(E13.5))
2 FORMAT(2(E13.5))
3 FORMAT(3(E13.5))
COMMON/MATIO/FORM, TOL
COMMON/HDRMPR/OSUB(2), KPAGE
COMMON/HEAD/TITLE(7), CASE(4), LINE
DATA RNUM1/21, 20, 19, 18, 17, 15, 12, 9, 8/
DATA RNUM2/39, 38, 37, 35, 34, 31, 30, 29, 28, 27, 26, 25, 23, 22, 21, 19,
~18, 17, 16, 15, 14, 13, 12, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1/
DATA NROWS/0, 0/
C:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C          PAGE HEADER SET UP
C:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
FORM=0.
TOL=0.
LINE=1
CASE(1)=LOCKHEED T
CASE(2)=TELESCOPE M
CASE(3)=MODEL (12 M
CASE(4)=ODES )
OSUB(1)=LGO HT SPE
OSUB(2)=CIFICATION
KPAGE=0
C:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C          INITIALIZE MATRICIES AND MODEL PARAMETERS
C:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
NM=44
NM2=2*NM
NEM=8
NESA=2*NEM
NU=21
NY=3
NZ=45
NO=24
NOA=NO
NOB=NO
NOD=NO
NOC=NY
NOJ=23

```

```

NRQ=NY
NRU=NZ
NRR=NU
NRX=NX
NRK=NX
NRP=NX
NRH=NZ
NU=NRU
CALL MZERO(AA,NX,NX,NRA)
CALL MZERO(BB,NX,NU,NRB)
CALL MZERO(CD,NX,NU,NRD)
CALL MZERO(CC,NY,NX,NRC)
CALL MZERO(MM,NZ,NX,NRM)
CALL MZERO(HH,NRW,NRW,NRW)
CALL MZERO(UU,I,NZ,NZ,NRU)
C:.....
C      SET UP AA MATRIX
C:.....
DO 40 I=1,NM,5
IF ((I+4).GT.NM) GO TO 40
M=I+4
READ(5,5)(WK(J),J=I,M)
40 CONTINUE
N=I-5
READ(5,5)(WK(J),J=N,NM)
DO 50 I=1,NM,5
IF ((I+4).GT.NM) GO TO 50
M=I+4
READ(5,5)(WK(J+NM),J=I,M)
50 CONTINUE
N=I-5
READ(5,5)(WK(J+NM),J=N,NM)
CALL IDENT(AA(1,NEMA+1),NEMA,NRA)
DO 100 I=1,NEMA
AA(NEMA+I,I)=WK(I+3)
AA(I+NEMA,I+NEMA)=WK(NM+3+I)
100 CONTINUE
AA(17,19)=1.
AA(18,20)=1.
AA(19,17)=-3947.8
AA(20,18)=-986.96
AA(19,19)=-.1257
AA(20,20)=-.0628
AA(22,24)=1.
AA(21,23)=1.
C:.....
C      SET UP BB MATRIX
C:.....
II=NM+NU
DO 300 I=1,II,5
M=I+4
IF (M.GT.II) GO TO 300
READ(5,5)(WK(K),K=I,M)
300 CONTINUE
I=I-5
READ(5,5)(WK(K),K=I,II)
DO 301 I=1,NEMA
DO 301 J=1,NU
BB(NEMA+I,J)=WK((I+2)*NU+J)
301 CONTINUE

```



```

      I=I-5
      READ(5,5)(WK(K),K=1,II)
      CALL MEQ(CC,MM,3,24,3,45)
      DO 210 I=1,NU
      DO 210 J=1,NEMA
      D=WK(44*(I-1)+3+J)
      MM(I+MY,J)=D
      MM(I+MY+NU,J+NEMA)=D
210  CONTINUE
      DO 211 I=1,NU
      DO 211 J=1,2
      D=WK(44*(I-1)+J)
      MM(I+MY,J+NESA+4)=D
      MM(I+MY+NU,J+NESA+6)=D
211  CONTINUE
C     CALL MPRINT(NRM,MM,NZ,NX,8,5,MM,0)
C:.....
C     SET UP UUI MATRIX
C:.....
      DO 212 I=1,NU
      UUI(NY+I,NY+I)=1.0E+06
      UUI(NY+NU+I,NY+NU+I)=1.0E+07
212  CONTINUE
      UUI(1,1)=1.0E+04
      UUI(2,2)=UUI(1,1)
      UUI(3,3)=1.0E+06
C     CALL MPRINT(NRU,UUI,NZ,NZ,8,5,UUI INUSE=0)
C:.....
C     SET UP UUI MATRIX
C:.....
      DO 213 I=1,NU
      UUI(I,I)=1
213  CONTINUE
      UUI(NU+1,NU+1)=3.95
      UUI(NU+2,NU+2)=3.95
C     CALL MPRINT(NRU,UUI,NRU,NRU,8,5,UUI,0)
C-----
C     SPECIFY OUTPUT AND CONTROL MAX VALUES
C-----
      DO 53 I=1,2
      SIG(I)=1.0E-07
53  CONTINUE
      SIG(3)=1.0E-12
      DO 52 I=1,NU
      MU(I)=1.0E-04
52  CONTINUE
      ITT=.FALSE.
      IF(.NOT.ITT) GO TO 444
      CALL MAREAD(NY,SIG,NY,1,MY MS UALS #)
      CALL MAREAD(NU,MU,NU,1,MU MS UALS #)
444  CONTINUE
C-----
C     ACTUATOR/SENSOR LABELING
C-----
      TITLU(1)=10H  FY1
      TITLU(2)=10H  FZ1
      TITLU(3)=10H  FZ2
      TITLU(4)=10H  FX3
      TITLU(5)=10H  FY3
      TITLU(6)=10H  FZ3

```

TITLU(7)=10H FZ4
TITLU(8)=10H FX5
TITLU(9)=10H FY5
TITLU(10)=10H FZ5
TITLU(11)=10H FZ6
TITLU(12)=10H FY7
TITLU(13)=10H FZ7
TITLU(14)=10H FZ8
TITLU(15)=10H FZ9
TITLU(16)=10H FZ10
TITLU(17)=10H FX11
TITLU(18)=10H FY11
TITLU(19)=10H FZ11
TITLU(20)=10H FY12
TITLU(21)=10H FZ12
TITLY(1)=10H LOS X
TITLY(2)=10H LOS Y
TITLY(3)=10H DEFOCUS
CALL MEQ(TITLY,TITLZ,3,1,3,45)
TITLZ(4)=10H Y1
TITLZ(5)=10H Z1
TITLZ(6)=10H Z2
TITLZ(7)=10H X3
TITLZ(8)=10H Y3
TITLZ(9)=10H Z3
TITLZ(10)=10H Z4
TITLZ(11)=10H X5
TITLZ(12)=10H Y5
TITLZ(13)=10H Z5
TITLZ(14)=10H Z6
TITLZ(15)=10H Y7
TITLZ(16)=10H Z7
TITLZ(17)=10H Z8
TITLZ(18)=10H Z9
TITLZ(19)=10H Z10
TITLZ(20)=10H X11
TITLZ(21)=10H Y11
TITLZ(22)=10H Z11
TITLZ(23)=10H Y12
TITLZ(24)=10H Z12
TITLZ(25)=10H LRY1
TITLZ(26)=10H LRZ1
TITLZ(27)=10H LRZ2
TITLZ(28)=10H LRX3
TITLZ(29)=10H LRY3
TITLZ(30)=10H LRZ3
TITLZ(31)=10H LRZ4
TITLZ(32)=10H LRX5
TITLZ(33)=10H LRY5
TITLZ(34)=10H LRZ5
TITLZ(35)=10H LRZ6
TITLZ(36)=10H LRY7
TITLZ(37)=10H LRZ7
TITLZ(38)=10H LRZ8
TITLZ(39)=10H LRZ9
TITLZ(40)=10H LRZ10
TITLZ(41)=10H LRX11
TITLZ(42)=10H LRY11
TITLZ(43)=10H LRZ11
TITLZ(44)=10H LRY12

VITA

VITA

Michael L. DeLorenzo is presently a Captain in the United States Air Force. He was born on April 2, 1952 to Joseph T. and Juanita K. DeLorenzo in Knoxville, Tennessee. In 1970, he graduated from Bearden High School in Knoxville, Tennessee and accepted an appointment to the United States Air Force Academy in Colorado. On June 5, 1974 he graduated from the Air Force Academy with a B.S. in Astronautical Engineering and Engineering Sciences and received a Second Lieutenant's Commission in the United States Air Force. From June 1974 to June 1978, he served as a Gyroscope Test Engineer for the 6585th, Test Group, Holloman Air Force Base, New Mexico. During that time he received an M.S. in Electrical Engineering from New Mexico State University. In 1978, he accepted a position on the faculty at the United States Air Force Academy in the Department of Astronautics and Computer Science. He served in that capacity until arriving at Purdue University in June 1980 to pursue a Doctor of Philosophy degree in Astronautical Engineering.

END

FILMED

1-84

DTIC